$12.00

ARRL AMATEUR RADIO

11th COMPUTER NETWORKING CONFERENCE

TEANECK, NEW JERSEY

November 7, 1992

ARRL AMATEUR RADIO

11th COMPUTER NETWORKING CONFERENCE

HOSTS: Radio Amateur Telecommunications Society (RATS)

American Radio Relay League

Printed in USA

$12.00 in USA

ISBN: 0-87259-406-8

First Edition

Foreword

The American Radio Relay League takes pride in publishing these papers of the Eleventh ARRL Amateur Radio Computer Networking Conference.

The papers this year reflect the growing exploitation of packet-radio technology by amateurs. In the development and use of new, more effective technologies and in the burgeoning list of applications to which the expanding networks are being put we see that packet is more than ever a key component of Amateur Radio. New developments include a high-speed link controller, improvements to the ROSE packet switch, experimental use of automatic link establishment (ALE) by amateurs, progress on the Clover II HF system, a new multimode controller card for the IBM PC and new packet software for the Apple Macintosh. On the applications front, packet-based telemetry, position reporting and advanced mail systems are among the advances reported in this year's papers.

As in previous conferences, all papers in these proceedings are unedited and solely the work of the authors.

David Sumner, K1ZZ
Executive Vice President

Newington, Connecticut
November, 1992

Table of Contents

Packet Radio at 19.2kB -- A Progress Report

by John Ackermann AG9V
john.ackermann@daytonOH.ncr.com
Miami Valley FM Association, Dayton, Ohio

Overview

This article briefly describes the technical aspects of the 19.2kB backbone and metropolitan area network (MAN) now operating in Southwestern Ohio.

Although there has been a lot of discussion of fast packet radio systems -- from 56kB on up to 10MB -- in the packet world, the fact remains that ten years ago we knew that 1200 baud systems were only a stopgap and that we needed to move to higher speeds before packet would come into its own. Today, there is still a dearth of equipment commercially available to operate at speeds greater than 1200 baud.

When Phil Anderson of Kantronics first started talking about his plan for a transceiver that would do 19.2kB out of the box, a group of hams here began to consider the possibility of building a high speed network to link the area's population centers, and to provide high speed user and server access as well.

The Kantronics equipment doesn't represent the state of the art in fast bits. But, it *is* the first system that allows a major improvement in packet performance using off-the-shelf components designed for the task. To put it simply, the people funding the Ohio network wouldn't have put up their money for the sort of do-it-yourself approach that other high speed packet systems have required.

Their goal was to build an infrastructure to support the general amateur community, not a plaything for tinkerers, and off-the-shelf equipment is what finally convinced the ham community that this system might actually work.

The clubs and individuals involved, the Miami Valley FM Association, the Central Ohio Packet Association, the Ohio Packet Council, K1LT, N8XX, and AG9V, ended up building a backbone linking Dayton, Cincinnati, and Columbus, Ohio through a total of five network switches. The backbone currently has about 150 miles of 19.2kB radio links and links to the existing Ohio 4.8kB network at Columbus.

Additionally, in Dayton the MVFMA and AG9V have built a MAN around a full-duplex repeater to provide a hidden-transmitter free network at 19.2kB for end users and servers. The MAN is tied to the backbone through a switch.[1]

RF Hardware

The Kantronics D4-10[2] is a UHF radio designed expressly for data communication services. It is rated at 10 watts output, though our tests show that most radios put out about 12 watts. It has two crystal controlled channels and by default comes from Kantronics with crystals for 430.55MHz installed. The receiver is a completely different design than the earlier 2 meter DVR2-2 radio, and has much better spurious signal rejection.

The radio has selectable receiver bandwidths, with a nominal 15kHz IF for voice and low speed use, and a 60kHz bandwidth for 19.2kB.

---

[1] I'll use the term "network" to refer generically to the backbone and MAN unless it is important to distinguish between them.

[2] Since this article deals almost exclusively with Kantronics equipment, I should note that none of the folks involved in this project are affiliated with that company.

The D4 supports normal analog signal inputs and outputs, but also has a TTL level I/O port which is intended for 19.2kB use. The TXD line on this port drives an FSK circuit that shifts the transmitted frequency +/- 10kHz from center for mark and space. The radio provides pulse shaping to meet FCC bandwidth requirements at 19.2kB, so the TXD signal need not be processed before feeding it to the D4.

The RXD line comes from the output of a comparator that acts as a data slicer driven by the discriminator. DCD is driven by the D4 squelch, and PTT is a normal ground-to-key signal.

Turnaround time is very fast. Using a squelch-derived DCD signal, two of the radios will talk to each other with a TXDelay of 5 milliseconds, though using 10ms or so probably is safer (these tests were done using the Ottawa PI card, which offers 1ms timer resolution).

We've had several of the radios in service for five or six months and have had no failures. We have seen some long-term frequency drift but nothing beyond what would be expected with 6 MHz crystals multiplied to 430MHz. Apart from one radio that has a serious stability problem over very small temperature excursions (we suspect either a bad component or a bad crystal), so far temperature stability has not been a problem.

A pair of D4s talking to each other over moderate paths will tolerate a bit more than 4kHz of frequency error. We recently learned that Kantronics has come up with a couple of modifications to both improve frequency stability and increase the amount of tolerable frequency error; with these mods the radios are supposed to tolerate about a 6kHz frequency error. We expect to have more information on these modifications soon and will be installing them in our radios.

The only adjustment we've found to be somewhat critical is the threshold setting for the RX data slicer. This adjustment sets the point at which the slicer shifts from outputting a 0 to a 1 and ideally should be set to cause that transition at the center of the channel.

We've seen a couple of radios where this adjustment (R17 -- a pot that sets the DC reference signal to the comparator) was out of adjustment. As a result, the radios wouldn't decode signals from an on-frequency transmitter, but would do just fine with a signal that was off-frequency in the direction of the threshold error. One of the mods referred to above supposedly reduces the touchiness of this setting.

Adjusting the threshold requires an on-frequency signal with tone modulation and an oscilloscope. Once the threshold is set, it seems to stay put. We think the problems we saw were due to initial misadjustment or to the banging around some of the radios have seen; none of the radios have required a second tweak.

The D4 works well in high RF environments. The MAN repeater antenna has several other commercial and amateur VHF/UHF systems in close proximity, and at his house Lew, K0RR, runs 100W on 435MHz (OSCAR uplink) with only four feet of separation from the Isopole that's hooked up to the D4 on 430.95. He receives packets without error during uplink activity.

RF Paths

We were a bit concerned about whether 10 watt radios operating in a 60kHz bandwidth would have enough power for the longish paths we needed to span. We have found that as long as the RF path is line-of-sight (or close to it), the D4's power is sufficient for paths of up to at least 50 miles.

But that "LOS" qualifier is critical. Even on relatively short (10 mile) paths, if the antennas aren't in the clear and above the foliage at both ends, the path won't work. This was driven home during our testing this spring, when AG9V was trying to access the MAN repeater, which was at an excellent location, but with the antenna only about 20 feet above ground.

AG9V is about 10 miles away and uses a 22 element K1FO beam up about 45 feet, but in a rather low and heavily forested area. Before

the foliage came out, the path worked moderately well. Once the leaves were on trees, however, signals *totally* disappeared. Raising the repeater antenna to its final height of 120 feet brought back rock-solid signals.

Most of our point-to-point links use Cushcraft 11 element 435MHz yagis on 5 foot booms. We've found that surplus 3/4 inch 75 ohm CATV hardline works well. The slight mismatch doesn't seem to cause a problem on either our simplex links or on the repeater antenna.

Modems

We are using two different (*very* different) modem schemes. The backbone, which presently uses Kantronics DataEngines running G8BPQ node software (see below) uses the Kantronics 19k2/9k6 modem. This is essentially a G3RUH design with doubled clock rate and all the analog parts bypassed; the output of the scrambler is fed into the D4 TTL port.

On the MAN, we're using a different approach. We are feeding HDLC frames directly into the D4's TTL port without using a scrambler or any other modem components. Essentially, we are treating the D4 as an RF modem. There have been many theoretical discussions about whether the scrambler is necessary to provide reliable clock recovery and demodulation of NRZI FSK signals, but our real-world experience indicates that using the K9NG/G3RUH scrambler as implemented in the Kantronics modem adds no significant benefit.

We have a 35 mile path operational without a modem, and it seems to be just as reliable as similar paths using the Kantronics modem. (It's been pointed out that the scrambler also allows use of a tracking data slicer, but since the Kantronics 19k2 modem uses the D4's internal slicer, that's not an issue here.)

Using the modemless[3] approach requires that DCD be derived from the D4's squelch. This isn't a performance problem, as the D4 squelch is very fast and solid. The 5ms turnaround time noted above includes squelch response time, so it's apparent that there's no speed penalty in this approach. I'll leave for another time the argument about whether a system should recognize only data, or any signal, on channel as DCD. Note, too, that because this configuration doesn't scramble the data, it will not talk to 'RUH modem-based systems.

Bit Bangers

Several different devices have been tested as packet assemblers and HDLC generators.

We have used the Kantronics DataEngine both with the 19k2 modem and our "modemless" configuration. We have several DEs in place running the G8BPQ network code and they do just fine supporting two 19.2kB links on the radio ports and a couple of slower links on the RS-232 port.

The DataEngine has one problem -- if it is connected to high-speed modems that allow the RXD line to chatter when there's no signal present, the interrupts generated will saturate the V40 processor and prevent it from properly servicing the serial port. This isn't very noticeable when running the standard TNC2 command set on the DE, but it is very apparent when running KISS mode, or using the G8BPQ node software.

Kantronics says they're working on a software fix for this problem. A relatively simple hardware mod can tremendously improve things -- simply gate the modem's RXD output with DCD so the chatter doesn't make it through to the DataEngine. We've made this change at our G8BPQ switch sites, and with it the DataEngine works well.

To use the DataEngine to feed HDLC directly to the D4, we use an internal jumper board (available from Kantronics for about $20) to provide the proper signals to the DE's radio

[3]Technically, the D4 is acting as an RF modem when it's being driven with digital signal levels. I use the term "modemless" to contrast with systems using a separate device between the output of the PAD and the input to the radio.

port. One bit of hacking is required: you need to add a 4020 or similar chip to divide the 16x clock the DE provides down to 1x clock, and feed that back to the TNC. The mod is no big deal; the chip fits neatly on the jumper board in "dead bug" style. Again, gating the RXD line will probably make the DataEngine happier.

A TNC with speeded up clock and components will also work well at 19.2kB. We've put three of PacComm's Tiny-2 TNC2 clones with 10MHz crystal and CPU[4] on line with no problems and full interoperability with our other HDLC generators.

By doubling the crystal speed, the baud rates are also doubled, so the 10MHz Tiny-2 will support 38.4kB on the serial port and 19.2 on the radio port without any further modifications. The interface to the D4's TTL port is nothing more than a cable from the appropriate pins on the modem disconnect header.

One minor annoyance is that the PTT line is not included on the Tiny-2's header. You can either hope that the SIO's RTS pin will sink the D4's PTT current, or pick up PTT from the Tiny-2 rear panel radio connector.

The Ottawa PI interface card is theoretically the best PAD to use at these speeds because its DMA data transfer can handle higher speeds with less CPU load than a PC serial port. K0RR and AG9V used PI cards for their first experiments with the D4 over short-haul links.

In the real world, the PI card works well for us, but in ping testing we see a troublesome loss of 5 to 10 percent of the packets sent even over near-perfect paths. We don't see this loss when using Tiny-2s. Dave Perry, VE3IFB, has revised the PI driver software for NOS, and that provided some improvement, but the problem still persists. This packet loss is almost certainly the result of some sort of timing glitch, but as yet we haven't tracked it down.

In short, we'd like to use PI cards in our NOS-based systems and switches, but we are being cautious until we fix the dropping pings. The Ottawa folks have been very helpful, and I'm sure we'll get this licked soon. (This problem is unique to our setup with the D4 radios; others are using the cards with the 56kB GRAPES modems and reporting excellent results.)

The other problem with the PI card is that it doesn't work with computers other than PCs, or (at least for now) software other than NOS.

In summary, at the moment the best price/performance ratio in bit generating hardware is the 10MHz Tiny-2 ($150) coupled with a 16550AFN UART in your serial card. The PI card ($120US) will be the best answer for PC NOS users once we get the timing bug worked out.

Repeater

The Dayton MAN is built around a full-duplex repeater. The repeater ensures that every station can hear every other station, and helps reliability by allowing use of high-gain yagis pointed at the repeater site.

The repeater hardware is nothing special; it's simply two D4 radios connected back to back through their TTL ports. A 4049 CMOS inverter provides buffering. The only trick we learned is that since the D4 uses op amps rather than true TTL devices to drive the TTL port, hooking a pair of radios directly back-to-back won't work. You need to use some judicious pull-ups and pull-downs to get proper signal levels.

The current controller has nothing more than the interface circuitry, a 15 second time-out timer, and a control function to drop the transmitter. The next generation will add appropriate ORing circuits so that a switch can use the repeater as a radio port. That will tie the repeater to the high-speed backbone.

We also want to add a bit regenerator to help the quality of the transmitted signal. We'll do this because it's the right thing to do, but at the

[4]The 10MHz "node version" of the Tiny-2 is available from PacComm for about $25 more than the standard model.

moment it doesn't appear that we are losing many frames due to distortion in the repeater.

The repeater uses no squelch tail; PTT is driven directly by DCD. The D4 squelch and keyup time are fast enough to permit a TXDelay of 15 to 20 milliseconds through the repeater.

The two D4 radios are connected to a small TxRx duplexer (four cavities with a little band-pass and a lot of notch filtering) and the duplexer feeds an 11.5dB gain Diamond fiberglass antenna through about 160 feet of 3/4 inch CATV hardline. The antenna is about 120 feet up at a site that towers (by as much as 100 feet) over the rugged Ohio skyline. Frequency separation between receive and transmit is 10MHz (420.95 in, 430.95 out) and we haven't noticed any desensitization.

There are currently six stations using the repeater, with paths ranging from about 1 mile up to 35 miles. The DataEngine, Tiny-2, and PI card are all being used to generate packets, and all three interoperate with no trouble. The repeater carries BBS, TCP/IP, NetRom, and PacketCluster traffic.

Switch Hardware and Software

At the moment, our nodes primarily use G8BPQ code running in DataEngines. We hope to change over to NOS-based switches that can handle both IP and NetRom switching. Johan Reinalda, WG7J, has NOS running on the DataEngine, and we plan to start experimenting with that code soon.

Apart from the inherent limitations of the NetRom protocol, the G8BQP/DataEngine combination works very well, and we've seen switches run for months without a reset.

Frequency Allocation

Finding space for a bunch of 100kHz wide channels on the 70cm band isn't easy. The ARRL-recommended segment of 430-431 MHz happens to fall in the top of the 426.75MHz ATV channel. Among other things, this means that an ATV transmitter puts its audio subcarrier at 430.75; a local ATV repeater uses that channel for its output and that rude surprise required us to change the repeater frequency shortly after we started testing

Apart from ATV, the 420-430 segment of the band is used in Ohio for voice links, many of which are uncoordinated.

After a lot of experimentation and negotiation (and with tremendous cooperation from DARA -- the Hamvention folks -- who agreed to move a bunch of their planned voice links for us) we wound up with the 420.5 to 421.0MHz range free for our repeater input and point-to-point paths. It appears that we can coexist with the ATV signal in the 430.5 to 431.0 segment if we stay clear of 430.75. We haven't tested yet, but we suspect that operation below 430.5MHz may result in interference with/to the video signal.

Kantronics ships the D4 by default with 430.55MHz crystals installed in channel 1; we got all our radios with that channel and use the second channel position for our operating channels. We don't have any links assigned to 430.55, so it is available for temporary and testing use and we can grab any two radios and make them talk to each other. We recommend that others follow this practice and reserve 430.55 as a testing and experimental channel..

The long and short of it is that frequency coordination will be a prime concern for users of D4 radios in populous areas. Living with ATV is likely to be the biggest challenge.

Acknowledgments

Primary funding for the backbone was provided by the Miami Valley FM Association, the Central Ohio Packet Association, the Ohio Packet Council, and Hank Greeb, N8XX. The Dayton MAN was funded by the MVFMA with minor contributions by AG9V. NCR Corporation kindly donated computer equipment that was surplus to them but critical to us; that equipment is be-

ing used to provide applications across the network.

Vic Kean, K1LT, is the technical guru for the project. He burned the ROMs, tuned the radios, arranged for the sites, and climbed the towers. Without Vic, this network would not exist.

Kantronics provided us with the pair of beta-version radios that got the whole project started, and with technical support along the way.

Finally, the support of K0RR, who bought the second PI card in town, and N8KZA, N8ACV, WB8GXB, and K8GKH, stalwarts of the MVFMA, was invaluable in getting the Dayton end of things running.

Telemetry Adapter for the TNC-2

By Bill Beech, NJ7P
P.O. Box 38
Sierra Vista, AZ. 85636-0038

Jack Taylor, N7OO
RR-2 Box 1640
Sierra Vista, AZ. 85635

Abstract

This paper describes a modification for the TNC-2 to allow 16 bits of digital I/O and 16 channels of analog to digital conversion.

1. Background

During the development of the TheNet 2.XX code, the need for control and monitoring of remote sites was a discussion topic between the authors. Kantronics had marketed the Weather Node and we were asked to provide an interface from TheNet to the Weather Node. The idea that there must be a more seamless solution drove the design of this adapter.

2. Design Alternatives

The Weather Node is an 8051 microprocessor based device to provide an ASCII interface to read the measured data. This design did not lend itself to interfacing with a node stack because the node stack uses a modified X.25 frame for internode communications on the RS-232 port. It did not provide the control functions necessary to provide remote control of the site.

The design discussed in this paper utilizes an 82C55 Parallel Interface Adapter (PIA) and an ADC0817 16-channel Analog Digital Converter (ADC). These IC's provide the necessary functionality to provide 16 bits of digital I/O, which are byte selectable as either input or output, and a 16 channel voltmeter with 20 millivolt resolution on the basic range of 0 to 5 volts. They are used in an adapter which fits in the TNC-2 and adds this functionality to the TNC-2.

3. Circuit Description

The adapter (see figure 1) gains its operating power and all but one required signal from its connection to the Z80 microprocessor. The adapter plugs into the Z80 socket, and the Z80 plugs into a socket on the adapter.

The 74HCT138 provides the address decode for the PIA and the ADC. The 74HCT02 provides the read/write qualification for the ADC and inverts the reset signal for the PIA.

The 8 bits of port A and B of the PIA are available for control. The ports can be set independently for input or output. Each bit represent a CMOS load as an input. Each bit can source/sink up to 2 milliamps as an output.

The lower 4 bits of port C of the PIA are used to select the ADC channel for conversion. The ADC is clocked from the 614 KHz signal on pin 5 of U4A in the TNC-2. (This is the only signal not present on the Z80). The conversion time is 100 microseconds. Each channel can have the basic range multiplied by insertion of a single resistor in place of the jumpers in the MULT headers, H2 and H3.

4. Software

The Telemetry Adapter software was incorporated in the TheNet Version 2.10 release. Since there is no generic AX.25 code for the TNC-2 in the public domain, this was the only software available for testing. As the adapter was to be used in a node stack, and the authors had experience with this code, this was not a problem.

The software allows reading the digital ports as hex bytes. The ports are written to in a decimal format. The voltmeter data is displayed as fixedpoint (i.e. 1.23). The software will allow integral multipliers of the basic 0 to 5 volt range (i.e. 0 to 20 volts with a 100k resistor) to match the multiplier resistors used on each channel. The ADC is switched through each channel continuously measuring the values. The software switches channels every 10 milliseconds.

The following exchange with a telemetry adapter equipped TNC-2 demonstrates the telemetry functions (**bold are user commands**, normal are tnc responses):

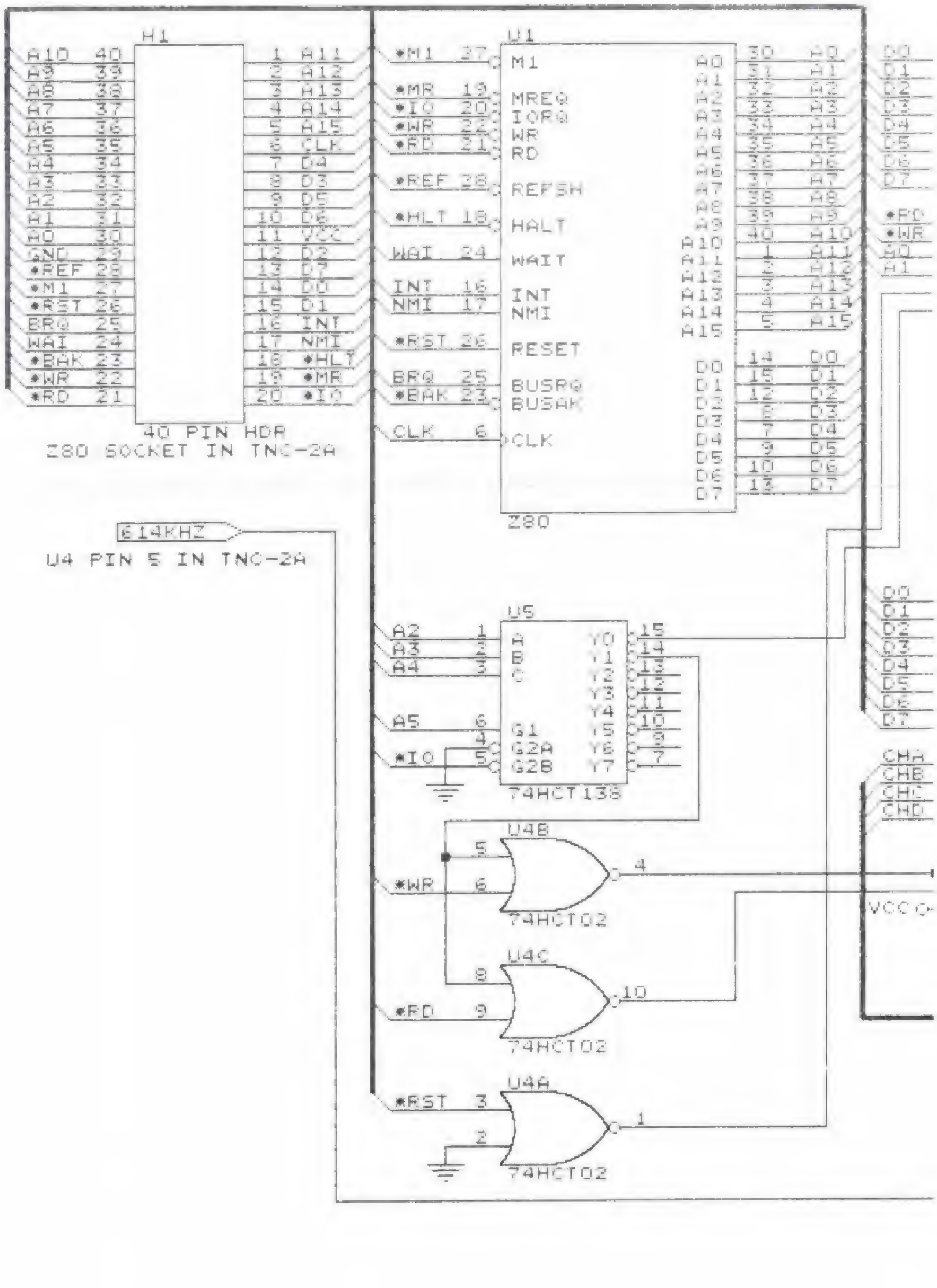
```
* c
CONN to NJ7P-4
t
SVATST:NJ7P-4} A=00 B=00
V0-7  4.99 3.46 2.35 1.62 1.09 0.78 0.50 0.37
V8-15 0.23 0.17 0.11 0.07 0.05 0.01 0.00 0.00
t 255 128
SVATST:NJ7P-4} A=FF B=80
V0-7  4.99 3.44 2.37 1.60 1.09 0.78 0.50 0.35
V8-15 0.25 0.15 0.11 0.07 0.03 0.03 0.01 0.00
t 128 0
SVATST:NJ7P-4} A=80 B=00
V0-7  4.99 3.44 2.35 1.62 1.09 0.74 0.50 0.37
V8-15 0.23 0.17 0.11 0.07 0.03 0.03 0.01 0.00
t 0 0
SVATST:NJ7P-4} A=00 B=00
V0-7  4.99 3.46 2.37 1.62 1.11 0.74 0.52 0.35
V8-15 0.25 0.15 0.11 0.07 0.03 0.01 0.01 0.00
b
DISC from NJ7P-4
```

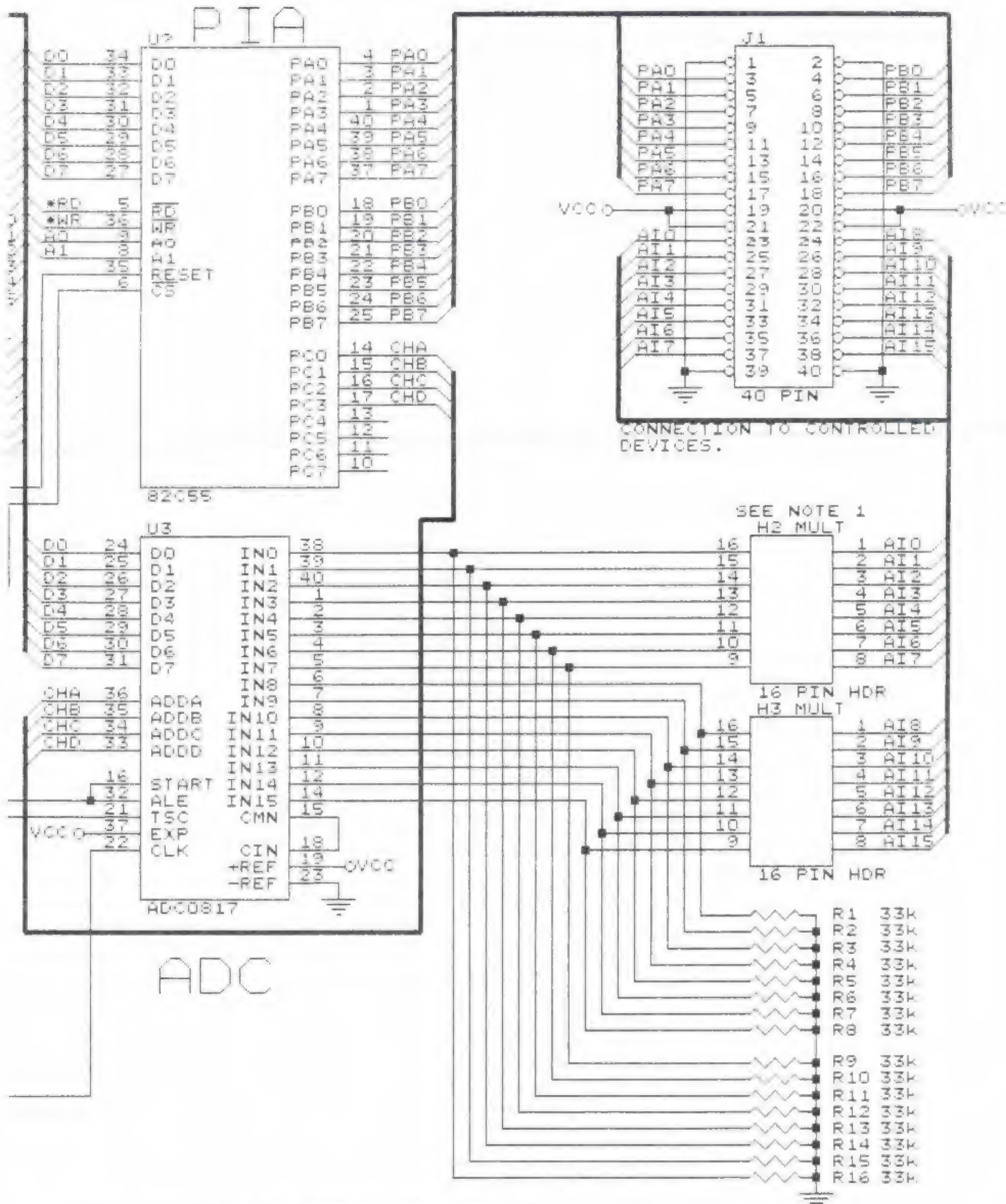
In this example both digital I/O ports are configured for output. Note that the commands for the bits are in decimal. The conversion routine for input uses decimal notation, and there is not enough room in the ROM for a hexadecimal routine. The voltages read here are from a resistor ladder network consisting of 16 4.7K resistors tying IN0 to IN15 together. IN0 is also connected to VCC and IN15 is connected to ground.

5. Applications

The obvious application would be remote monitoring of power supply voltages and transmitter forward and reflected power. Power supply and battery voltages could be measured by putting a 100K resistor in the multiplier, and setting the software multiplier to 4, yielding a voltmeter range of 0 to 20 volts with 80 millivolt steps. RF power output could be measured by setting the multiplier and multiplier resistor for an appropriate range. The Radio Shack 10k thermister, PN 271-110, will read temperatures between -50 and 140 degrees farenheit in a non-linear fashion.

Applications providing lower voltages could use an operational amplifier to bring the range up to that of the basic 0 to 5 volts. This was done with the homemade anemometer based on a 99 cent Radio Shack permanent magnet


H1
40 PIN HDR
Z80 SOCKET IN TNC-2A
614KHZ
U4 PIN 5 IN TNC-2A
U1
Z80
U5
74HCT138
U4B
74HCT02
U4C
74HCT02
U4A
74HCT02
VCC



NOTE 1: MULTIPLIER RESISTORS WORK IN
IN CONJUNCTION WITH R1-16 TO
CHANGE THE BASIC 0-5VOLT RANGE
I.E. A WIRE IN THE MULT
POSITION IS 0-5VOLTS, AND
A 68K RESISTOR IS 0-15VOLTS.

motor. This anemometer provided 0.075 volts at 50 miles per hour and was quite linear.

6. Conclusions and Future Directions

The adapter has functioned flawlessly for many months. There is a lot of work to be done on interfaces for it to sense and control the real world.

The ability to pass the measured values to a remote collection station automatically would be nice. With the code constraints, it might be better to have a control node query data from various remote sites and inform the system operators whenever a problem is detected.

7. References

Intel, "Peripherals", 1990

MFJ ENTERPRISES, "Model MFJ-1270B/1274 Packet Radio Terminal Node Controller TNC 2, Rev 3.1", First Edition, 1986

National Semiconductor, "Data Acquisition Handbook", 1978

AUTOMATIC AX.25 POSITION AND STATUS REPORTING

Bob Bruninga WB4APR
115 Old Farm Court
Glen Burnie, MD 21060

For the last two years the Naval Academy has used a packet radio network for communications with its boats during summer cruises. The packet radio system not only provides the connectivity typical in AX.25 radio networks for the exchange of messages, but the automatic beacons from the afloat units provide near realtime position reporting of the units at sea. The purpose of this article is to describe the Academy system, particularly the use of beacons for position and status reporting and to suggest the advanatages of such a system for use in emergency situations and network management in other AX.25 packet systems. Detailed formats for automatic position and status reporting are provided. In any communication network for any purpose, station location and status reporting are at least the second most important function, if not the first.

PACKET RADIO TACTICAL COMMUNICATION NETWORK

The objective of the AX.25 network at the Naval Academy is to provide position and status reporting and to permit the exchange of record traffic between the Academy and its fleet of almost forty Yard Patrol craft (YP's) and sailboats. The exchange of message traffic is straight forward using the internal PBBS of one Kantronics KAM, linked by Kantronics KA nodes to three HF channels, one VHF frequency for local operations, and one UHF SATCOM channel. (A description of the satellite portion of this network were published in the 1992 AMSAT Technical Symposium in October 1992) The innovation in the network, is the use of periodic packet beacons for position and status reporting. Central to the success of this beacon system is a special program which monitors the packet channels and accumulates the position and status beacons and then provides a tactical color map display of the location of all units on a scaleable map of the East Coast.

HARDWARE IMPLEMENTATION

For reliability, computers were initially avoided in the system. The master station first consisted of three dual port TNC's, one on each HF frequency. The audio from the VHF ports of these three TNC's are all summed together and fed to the SATCOM transceiver and local VHF transceiver as shown in figure 1. This simple audio node serves as a local area network to link everything together without additional hardware and node complexities. The KA node function built into the dual port TNC's manages traffic between the TNC's. The message mailbox system is simply the 16K PBBS internal to the one TNC on the 6 MHz frequency. This PBBS can be accessed directly from HF on 6 MHz, from VHF, and from the SATCOM system via the audio node. On the two other HF frequencies, the boats must first connect to the KA node TNC on that frequency, and cross link to the PBBS (via the audio node). Kantronics KAMS were used initially for their dual port and KA-node capability at the master station; but because the KAMS have no way to manually initiate the forwarding of a message from one PBBS to another, we are now buying MFJ's and PACCOM TNC's for the remaining boats.

The master station is further available to officers and staff via the Academy local area network (LAN) which provides a serial port in every office at the Academy. Again, without complexity, this was accomplished simply by connecting the TNC RS-232 serial port to the network and programming the network to recognize that address as a Host. This LAN interface allows duty officers to log into the TNC PBBS from anywhere in the Academy as well as dial-up from home to list and read message traffic. A one-transistor interface between the "CONNECTED" LED on the TNC and the LAN handles the "ready" and "busy" handshaking. Three similar TNC interfaces permit HAMS on the Naval Academy LAN to access MIR, DOVE and the local packet network.

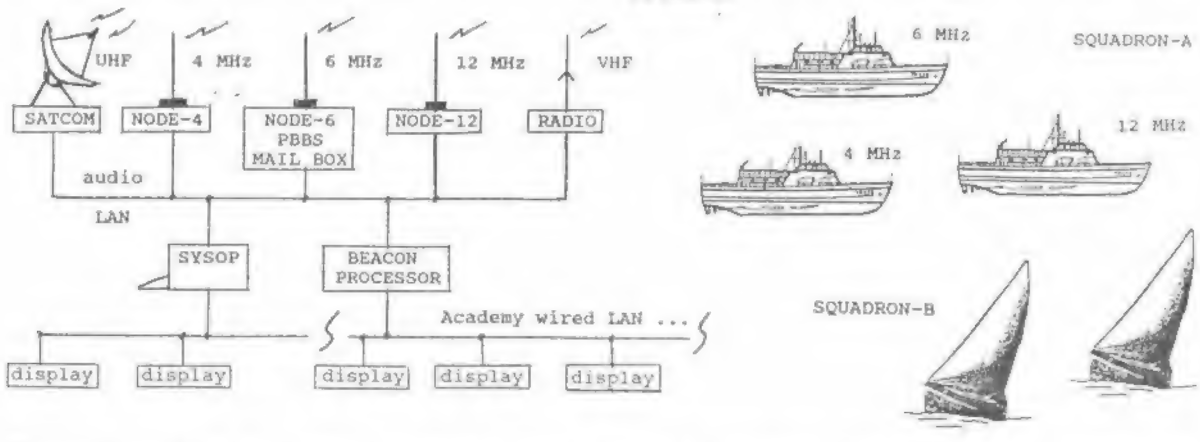


Figure 1. The Academy Network consists of three HF nodes integrated with a local VHF frequency and UHF SATCOM system via dual-port KA nodes. By August 92, 10 boats were configured with packet radio.

POSITION REPORTING BEACONS

To take advantage of propagation openings on HF, all units in the network are programmed to transmit a periodic beacon signal once every ten minutes. For the boats, the beacon is loaded with position and status. For the master station at the Academy, the beacons include short announcements or lists of traffic pending. On the boats, a casual glance at the time of receipt of the last beacons on the CRT terminal indicates propagation conditions. If the last beacons were received in the last 10 minutes, then conditions are probably good for message traffic.

At the master station during the first summer, all position report beacons from the boats were logged on a printer. The beacons were foramtted so that position, course, speed, fuel, water, casualties, next port, estimated time of arrival, and intentions were all included in the single line beacon text. By the second summer a computer program was written to parse and sort all beacons and display the position and movement of the boats in full color graphics which included charts of the entire East coast with user selected scales of 4 to 1024 miles.

To be sure that beacons on all frequencies are visible to all users, particular attention was made to the setup parameters in each TNC. First, all TNC's on the boats are set to beacon via the address ECHO. Then the gateway callsign of the three dual port TNC's at the Academy are programmed with this callsign of ECHO. This way, a beacon originating on any one HF frequency is digipeated (ECHOed) onto the master station audio node which in turn goes out on VHF and SATCOM. Any beacon originated on VHF or SATCOM is similarly digipeated (ECHOed) out on all three HF frequencies. A Beacon can be further distributed from one HF frequency onto the local VHF node and then back out on the other two HF frequencies by simply beaconing via ECHO, ECHO.

POSITION and STATUS DISPLAY SOFTWARE

The key to the success of the packet reporting system is the position and status display software which provides everyone with fresh graphic display information on the position and status of all units. The use of computer displays in real time is far superior to the colored pins on a wall chart which had been used for position display in the past. The tactical display using EGA graphics shows a map of the East Coast at any scale between 4 to 1024 mile range centered anywhere from Nova Scotia to the Florida Keys. A picture of the display showing several units enroute to New York is shown in figure 2. Using the cursor, any unit may be selected for a detailed display of data on that unit. Further, a single key stroke will dead reckon all unit positions to their estimated current positions based on course and speed from their last reported

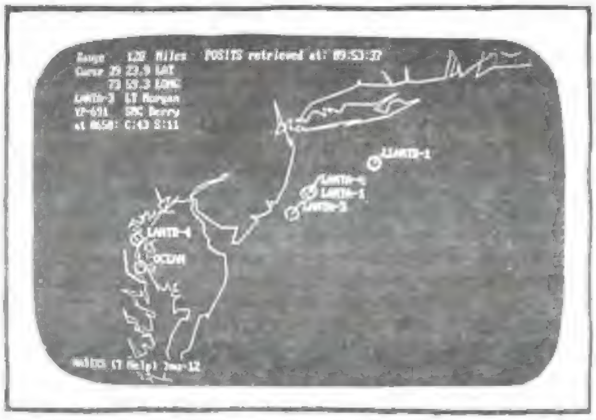
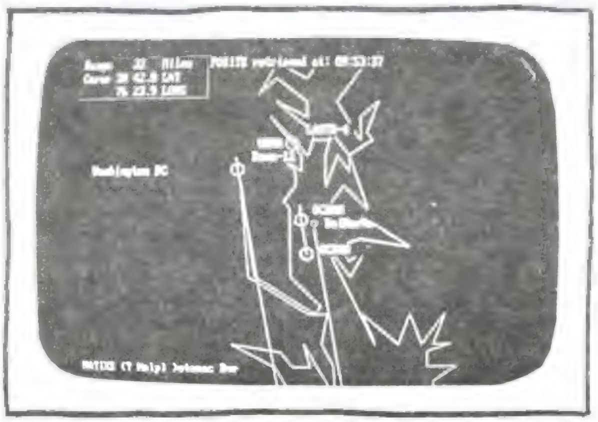


Figure 2. The tracking and display software can display the location of all packet units to any scale from 1024 down to 4 miles. Here, several units are enroute to New York.



position. It is this tactical display software which would be useful in other AX.25 packet radio networks for displaying the location and status of all participating stations.

REMOTE TELEMETRY VIA PACKET

An interesting sidelight to the packet network was the use of a Kantronics Telemetry Unit on one boat to monitor seven channels of engine conditions remotely and dump these readings over the packet radio link. Unfortunately the only way to locate engine readings of interest was to transmit every single data sample (16K per day) over packet and then search for the few high and low readings of interest. The result was a thousand-to-one ineffeciency in the use of the packet channel. We were disappointed in the absence of any search, max, min, rate, or threshold comparison commands in the KTU to help identify high or low readings of interest prior to transmission. Secondly, there was no way to remotely select the KTU. The KTU must have a dedicated TNC or a human operator to manually switch it into the circuit. For these reasons, we found the KTU to be of no value as a remote telemetry unit in this application.

HISTOGRAM OF BEACONS PER HOUR for 07-26-92. (Usually transmitting 6 per hour)
SSID shows assigned frequency of operation for that unit.

Time(EDT)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	23	22	21	20	19	18	17
LANTA-6	.	.	1	2	3	2	4	4	.	.	.	.	.	.	1	.	.	1	1	2	.	.	.
LANTA-12	.	1	1	2	3	2	5	3	2	3	2	.	.	.	.	1	.	.	2	4	1	3	.
LANTA-4	3	1	.	.	2	1	2	1	3	.	.	.	.	.	.	.	.	.	.	.	.	3	3
OCEAN-6	1	2	1	2	2	1	2	1	2	2	2	2	1	2	1	2	2	2	2	2	2	2	2
LANTB-6	4	1	4	2	3	3	5	4	2	2	.	.	1	2	.	1	1	1	1	2	3	2	2
LANTB-12	2	4	5	6	4	6	6	6	6	3	.	.	2	2	.	3	9	6	3	5	5	2	3
LANTB-4	.	.	.	.	1	3	4	.	.	.	.	.	.	.	.	.	1	.	.	3	.	.	.
BRMUDA-6	4	6	2	2	6	3	1	2	.	.	.	.	.	.	.	.	.	1	.	.	1	1	2
NEWENG-6	3	4	3	6	6	5	3	2	.	.	.	.	.	.	.	.	.	.	.	1	.	3	.

NOTES: LANTA arrived NY this morning. LANTB arrived Newport RI last night. Sailboats enroute up Deleware coast and into Deleware Bay.

Figure 3 Statistics on the number of beacons received per hour from each unit over the last 24 hours are available to system users with a single keystroke.

OPERATIONS

Beacons from the deployed boats are automatic as long as the packet terminal is on and the radios are properly configured. The crews are encouraged to update their beacons once a watch every four hours or whenever significant changes occur. Similarly, the crews are encouraged to watch for beacons from the master station and to check in at least twice a day when conditions are favorable. Each squadron has three boats equipped with packet, one on each of the three HF frequencies at 4, 6 and 12 MHz to take advantage of frequency diversity. The system software accumulates statistics on the number of beacons received from each unit per hour and makes that information available to users as shown in figure 3. Throughout the summer on HF, 40 percent of all beacons were received within the first ten minutes, 80 percent within the first hour, and 90 percent within 4 hours as shown in figure 4.

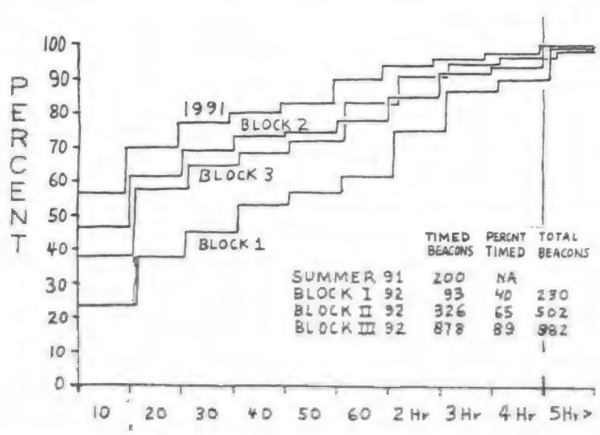


Figure 4. ELAPSED TIME BEFORE RECEIPT

RESULTS

Although the Naval Academy system was only experimental the first summer with three boats, more than 400 messages were exchanged over the system and 200 position reports were logged. Communication success measured by the number of successful communication events per four-hour period with packet approached 83% compared to about 20% for the twice per day HF voice system used before. The success of the HF beacons and our very low priority on gaining Navy UHF satellite time, showed the advantage of having a fully integrated HF system as part of the UHF satellite AX.25 network.

By the second year with ten boats configured for packet, there were always three boats per squadron able to monitor the three different HF frequencies for frequency diversity. Over 600 messages and 1700 position reports were logged. Further, the tactical display software transformed the system from an interesting experiment to a useful and productive tool in time for most of the summer of 1992.

AUTOMATIC NAVIGATION-TO-TNC INTERFACE

A highlight of the summer was the three week cruise of the one boat that is configured as an oceanographic research vessel. With GPS and several computers on board, a serial port was dedicated to send a new GPS position report beacon text to the TNC every ten minutes. At a cruise speed of 10 knots, this resulted in a fresh position report accurate to about 50 feet every two miles during a 300 mile cruise. A replay of the positions of this boat are shown in figure 5. This fall, we will be working on a direct interface between a GPS receiver and the TNC without any need for the computer.

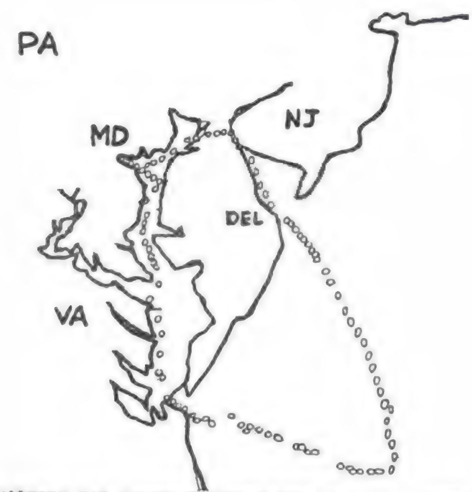


Figure 5 A plot of the 3 week track history of the one unit equipped with an interface to a GPS receiver to provide an automatic fresh position into the TNC beacon text every 10 minutes.

The National Marine Electronics Association (NMEA) 0183 standard defines a 4800 baud serial data position reporting format for interfacing commercial navigation equipment. By simply programming a TNC for 4800 baud and placing it in the UNPROTO CONVERSE mode, nothing else is needed for AX.25 position reporting via radio as shown in figure 6. Unfortunately, however, there are two problems which must be resolved before this connection is viable.

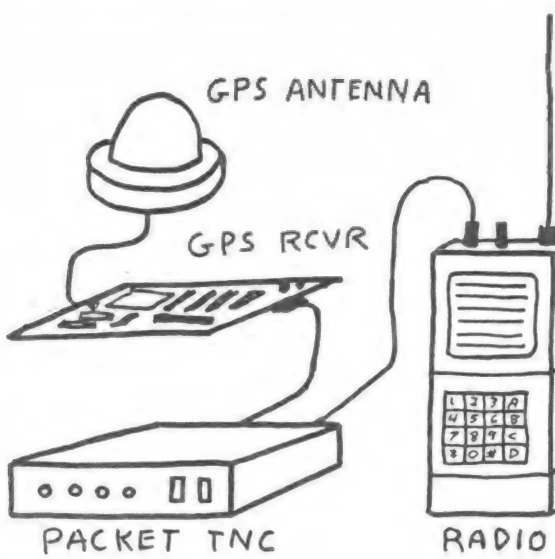


Figure 6. A complete automatic position reporting system can be assembled using a GPS receiver interfaced to a radio using an AX.25 packet radio TNC.

First, the NMEA 0183 interfaces on all GPS receivers we studied repeat a series of over a dozen lines of navigation data at a period of once every 2 seconds or so. We only need the one line which contains LAT/LONG; and we only need it once every 10 minutes or so. Manufacturers need to provide a user programmable periodicity for their existing formats. One company, MAGELLIAN, makes a GPS receiver prototyping kit ($550) which implements a user definable periodicity from 1 second to 5 minutes, which is useable for AX.25 applications.

Second, the standard TAPR-2 TNC does not include a provision for powering up in the UNPROTO-CONVERSE mode (the PK-232 does). Without this power up feature, a terminal and human operator, or PC, is also required to place the TNC in UNPROTO-CONVERSE mode with the "conv" command. We are proposing to the TNC manufacturers to provide a power up UNPROTO-CONVERSE mode as follows:

UNUP (ON/OFF) - UNPROTO CONVERSE on Power UP
- ON - The TNC will power up in UNPRO-CONV mode. From then, all operations are normal, ie: a ctrl-C will return you to command mode.
- OFF - Default setting. Normal TNC modes.

UNPERM (ON/OFF) - UNPROTO-CONVERSE PERMANENT
- ON - Same as UNUP except that the TNC, if left in cmd: mode will always revert back to UNPROTO-CONVERSE after a set time period. That time period is set using the existing CHECK parameter.
- OFF - Default setting. Normal TNC modes.

Once these two problems are resolved, automatic vehicle location using AX.25 can be implemented at a surprisingly low cost, by simply plugging in a navigation device to your existing mobile packet system. LORAN-C devices with NMEA-0183 interfaces can be purchased for under $400 from any Marine Electronics store. GPS receivers are not far behind at around $900 and falling rapidly! At least one LORAN receiver, the NORTHSTAR 800 already has the mods noted above. Unfortunately, however, we cannot use LORAN at the Naval Academy because we are located only 1 mile from the US NAVY Megawatt VLF transmitting station which wipes out LORAN for about 20 miles around us.

Even without the two modifications noted above, automatic vehicle location is still very easy to implement using a simple PC program between the navigation device and TNC to properly configure each device at power up, and to reduce the NMEA-0183 reporting periodicity to something reasonable for the radio channel. Still, the cost of implementing automatic vehicle location via AX.25 for under $600 is far less than the cost of commercial systems which have been under development for over 20 years.

AUTOMATED POSITION REPORTING SYSTEM (APRS)

The tactical display software used in the Naval Academy application could be very useful in other packet radio networks, especially in support of emergency and disaster communications, or for tracking recreational vehicles, mobile HF'ers, or offshore boaters. There is even an air traffic control net on 14,278 KHz which could use it to track air contacts. A single VHF frequency could be used to track all local boaters in the Chesapeake Bay.

In current AX.25 packet radio networks, the Monitor Heard log in each TNC is the only function which provides realtime information about activity on the network. This information only identifies the most recent callsigns and when they were last heard. The APR software which evolved from the Academy network is a dramatic improvement on this concept, as it displays all callsigns geographically when heard. Different colors show activity on different frequencies. A glance at the screen shows which stations are currently active. Lines can be drawn between pairs of connected stations including all intermediate links. In this way, geographically correct network charts just appear on the screen after monitoring the channel for only a few minutes of heavy use.

The key to this system, of course, is knowledge of every station's geographical position. Using beacons, as we have done at the Naval Academy, is the mechanism. Periodic beacons are not required for fixed stations. In fact, as more and more stations run this software and collect position reports from cooperative stations, a community file of station locations can be assembled and distributed via BBS files! If the idea cought on, BBS's could actually monitor beacons and collect and maintain the position files. Local stations need only update their position data when they move or go portable! Then, only one properly formatted beacon need be transmitted successfully for the community to grab your change in status.

Going back to the monitor heard logs, TNC firmware could be modified to save the text of the latest beacon from each station as well as the time of receipt. A new TNC command would give the owner the option of collecting only beacons and beacon text in his MH log, or not.

APR SYSTEM BEACON FORMATS

Since there are many different applications for this position reporting system, the APR software recognizes several formats of position and status reporting. Table 7 shows the basic formats which include provisions for both automatic and manual reporting in several coordinate systems. Each field in these formats is fixed length except for the status and comment field which is used to report a variety of status attributes.

TABLE 7. BEACON REPORTING FORMATS

Each field is fixed length except for the status and comments. The examples represent a beacon transmitted at 1450 on the 29th day of the month at a position of 38 degrees 45.54 minutes North, 76 degrees 29.43 minutes West, on a course of 038 degrees at 13 knots on a frequency of 145.05 MHz. The status characters indicate that the station is an ARES member with emergency power operating marine mobile capable of operating on all HF bands, plus mode A satellite, with all mode capability on 2 meters and 70 cm FM. He can also operate RTTY, ATV and Marine VHF. The definition of these status characters are given on the next page.

MOBILE PLATFORM WITH COMMENTS: Day,time,lat,long,course,speed,status

```
@291450/+3845.5/-07629.4/038/013\SAEh*nS1U2V4W/comments to end of line
```

NMEA-0183 LORAN/GPS INTERFACE: Identifier,latitude,longitude

```
$GPGLL 3845.54,N,07629.43,W   (from a GPS device)
$LCGLL 3845.54,N,07629.43,W   (from a LORAN-C device)
```

FIXED STATION INITIAL REPORT: Latitude,longitude,frequency,status,comments

```
#+3845.5/-07629.4/145050\SAEh*nS1U2V4W\comments to end of line
```

FIXED STATION UPDATE: frequency,status,comments

```
&145050\SAEh*nS1U2V4W/comments
```

N-S-E-W REFERENCE: Day,time,N/S and E/W offsets from reference point

```
$M291450/+06.3/-38.7/Baltimore\SAEh*nS1U2V4W\comments  (in Nautical Miles)
$K291450/+10.8/-34.2/Annapolis\SAEh*nS1U2V4W\comments  (in Kilometers)
```

STATUS FORMAT DEFINITIONS

The status block is a variable length character block which begins with a backslash (\) and goes to the end of the beacon. It contains a collection of alpha-numeric characters each of which represents a special attribute. The first character defines which table of definitions will be used for the attributes and the remaining characters represent the data. The first table that I have defined is the station (S) table for representing station information. Up to 60 additional status tables may be defined using the upper and lower case letters and numerals. Only the numerals are position dependent and are used to amplify other categories such as for H, h, U, V, or S. In this table, U distinguishes all-mode capability from V which assumes FM. General comments may be included at the end of the block following a slash (/) character.

Table Name: STATION ATTRIBUTES
Table identification character: S

A - ARES	a - aeronautical
B - Beams	b - battery pwr
C - CAP	c - cw
D - DX	d - digipeater
E - Emer-power	e - EMT trained
F - Field Portable	f - foxhunter
G - Generator	g - gps equip
H - HF with linear	h - hf other
I - Intl Marine VHF	i - inmotion
J - TBD	j - TBD
K - APLINK	k - TBD
L - Landline modem	l - loran equip
M - MARS	m - mobile
N - Node	n - nautical/marine
O - AMTOR	o - optical
P - PBBS	p - phonepatch
Q - ARRL mbr (QST)	q - cq
R - Races	r - repeater
S - Satellite	s - sstv
T - TCP-IP	t - rtty
U - USB,LSB & CW	u - microcomputer
V - VHF	v - ATV (FSTV)
W - Wx/Marine band	w - weather equip
X - 10 GHz and up	x - fax
Y - TBD	y - TBD
Z - TBD	z - TBD

1 -160 meters	1 - 50 MHz	1 - Mode A
2 - 80 meters	2 - 144	2 - Mode B
3 - 40 meters	3 - 220	3 - Mode J
4 - 30 meters	4 - 440	4 - Mode L
5 20 meters	5 - 902	5 - Mode S
6 - 18 meters	6 - 1296	6 - TBD
7 - 15 meters	7 - 2250	7 - UOSAT
8 - 12 meters	8 - 3230	8 - PACSAT
9 - 11 meters	9 - 5650	9 - 9600 ax25
0 - 10 meters	0 - scanner	0 - 1200 ax25
* - All/most	* - All/most	* - All/most

A second table of definitions for emergency and disaster communications is suggested below: ALthough the (S)tation table shown above is reasonably complete, the Emergency table is only offered as an example. I hope the experts in the field of emergency communications such as ARES and RACES will take it upon themselves to establish and maintain their own definitions.

Table Name: Emergency operations
Table identification character: E

A - ARES	a - ambulance
B - BEDS # follows	b - bilogical teams
C - CAP	c - city
D - Doctor	d - disaster team
E - Emer-power	e - EMT trained
F - Fire station	f - food
G - Generator	g - government (fed)
H - Hospital	h - helo
I - Intl Marine VHF	i - inmotion
J - TBD	j - TBD
K - TBD	k - TBD
L - Landline modem	l - loran
M - MARS (military)	m - mobile
N - Nuclear equip	n - nautical/marine
O - AMTOR	o - oil spill equip
P - Police	p - phonepatch
Q - Headquarters	q - freQs #'s follow
R - Races	r - RedCross
S - Shock-trauma	s - state
T - TV/radio/press	t - triage
U - Utility	u - microcomputer
V - Video ATV	v - vehicles # follow
W - Weather station	w - water
X - Xray capability	x - fax
Y - TBD	y - countY
Z - TBD	z - TBD

1 - 144 MHz	1 - Ladder truck
2 - 220 MHz	2 - Pumper truck
3 - Scanner	3 - Hazardous material
4 - 440 MHz	4 - Ambulance
5 - HF	5 - water truck
6 - Marine VHF	6 - bus
7 - Other govt	7 - Earth moving
8 - Cellular	8 - communications
9 - wormhole	9 - food delivery
0 - CB	0 -

SUMMARY

The purpose of this article is to use the example of the Naval Academy AX.25 packet radio network to show the value and capabilities of a system which includes position and status information in packet beacons. The extension of this capability into other AX.25 packet radio networks could have many benefits. A previous article on the subject of position and status reporting in packet radio networks appeared in the February 1991 issue of Gateway. I intended to evolve the APR software from the Naval Academy software, but by the end of this summer's intensive use, there were enough lessons learned to require a significant re-write before the system could be extended to world wide use. The Latitude and Longitude fields needed to be extended for whole Earth coverage, and the time fields needed to be extended beyond 24 hours to include the day as well. Also, provision for other beacon reporting formats including the NMEA 0183 format was required. Other formats tailored to specific applications, such as emergency operations and statute mile position references were suggested in the earlier Gateway article and are included here. I must emphasize that APR software for the general AX.25 community is NOT YET COMPLETE. As I reorganize the formats of the Academy System for next summer, however, the necessary changes will be included for a more general application.

Fast Amateur Link Controller - an allround solution using a SCSI bus

Werner Cornelius, DG3DBI @ DB0IZ.DEU.EU, Am Siepen 17, D-4630 Bochum 1

Two years ago fastening of interlinks and user entries called into being the project FALCon (Fast Amateur Link Controller). Final aim of this project is to develop hard- and software conceptions answering new and wider requirements (higher baud rates for example). The project was initiated by DG3DBI aiming at the development of new and powerful hardware for digipeaters and terminal node controllers. Exchanging ideas with Thomas, DL1EBQ, sysop of the German digipeater DB0ME was followed by Thomas offer to test newly developed hardware at DB0ME. Because of lack of time the software development was given to Walter, DG9EP, an experienced programmer, who set up a wholly new software for use at digipeaters - DigiWare. Following, development and fundamental characteristics of the hardware - FALCon - are described.

To cover all requirements, several test configurations had to be assembled. For this work was carried out only during leisure time, development took about two years.

Requirements to a new hardware

What features a new hardware has to show to match present requirements and also give space to further demands ?

First aim was to create a general conception for use as digipeater as well as terminal node controller. By change of its software, a FALCon can be used both at home and at a node.

Second, a common available standard microprocessor should be chosen to provide software development with software packages and computer systems widely spread among radio amateurs.

To meet higher data flux, the controller must have enough memory. This memory should be equipped with a battery backup to save parameters and software permanently.

An analysis of conventional solutions had the result that higher baud rates require a better handling procedure for packet data. Usual methods of processing receive and transmit data are polling and interrupt service routines for the serial ports. These ineffective methods should be replaced by consequent use of direct memory access by means of hardware DMA controllers.

Each controller should be provided with universal interfaces allowing both high speed communication between the controllers and external computers when needed. Though, the hardware conception of FALCon includes no homemade and self-defined interfaces - only standard interfaces are used that are available on wide-spread computer systems.

Other possible items were the provision of additional features as a real-time clock and digital I/O ports to control external functions.

The development steps of FALCon

As a conclusion of the forgoing stated requirements several test boards were assembled in wire-wrapping technique. These first test units were still mounted on two or three printed circuit boards connected with a bus system.

In the first versions of FALCon a NEC V50 processor was used with a clock frequency of 8 MHz [1]. In this connection, a very interesting fact is that FALCon was working in this configuration yet before Kantronics delivered their Data Engine, its heart being a smaller version of the V50, a V40 processor.
The early FALCon was not equipped with special interfaces for internal communication between the controllers. For diagnostic purposes, there only was a V24 port. The main memory was provided with dynamic RAMs due to lower price and lower need of room. Test software was developped externally on an IBM compatible PC and then loaded into an EPROM simulator substituting FALCon s EPROMs. Already this early version had four HDLC/SDLC interfaces integrated in two special chips. These integrated circuits are compatible to the well known SCCs of the type 8530. Additionally, the ICs provide two DMA blocks and an enhanced FIFO (first in, first out) per channel allowing much faster I/O than the conventional 8530. This conception of driving the HDLC/SDLC ports was proofed in practical operation at the net node DB0ME and was kept troughout further development of FALCon.
After optimizing these items the connection between several controllers was carried out. For realisation, several standard interface conception are available, which can be divided into two groups:

a) Serial interfaces
Serial ports benefit from having only few connecting lines between devices. Hence, the devices can easily be galvanically seperated. A disadvantage is, that a high effort is required for the regeneration of clock pulses.

b) Parallel interfaces
In comparison to serial ports, parallel interfaces have the serious disadvantage of using more transmission lines but can provide for high data transition rates. Galvanic separation is difficult to achieve.

From the point of view of an amateur radio station, galvanic separation is not required in most of the cases and mostly even not possible because single components are already connected with each other due to power and ground lines. The effort necessary to make decoupling perfect is out of all proportion when looking at the achieved advantage.
Correspondingly, parallel and serial interfaces could be treated uniformly in this application. Analysed werde three serial (V24, Ethernet, Arcnet) as well as two parallel interfaces (IEC bus, SCSI bus).
It was found to be that between the serial interfaces Ethernet and Arcnet require expensive components and need too much space on the pcb. To meet the specifications, serial interfaces have to be separated galvanically in any case. Because of this, usage of Ether- and Arcnet was rejected, although a test facility already had been realized. The V24 interface is the cheapest and most easy way of connecting two devices. Of great disadvantage of course is the low data transfer rate and the fact that only two devices can be coupled together.

However, as any personal computer (i. e. IBM PC, Atari) provides a V24-interface, at least one interface of this type should be available.
There are several industrial designs for VLSI chips available, which simplify implementation of parallel interfaces. Among them, there are the IEC- and the SCSI bus. The IEC bus normally is used for connection of measuring instruments but requires a specific and very expensive connection cable between the devices. Using a SCSI bus a simple flat cable is needed. Additionally, a component was found that integrates a complete SCSI interface together with DMA drivers in one single chip. In comparison with it, other well-known IEC bus components need external logic and drivers. Accordingly, the SCSI bus was chosen which has counterparts in many computer systems used by radio amateurs.

Design integration

After basic circuit design and test attempts have been made to integrate the complete system on a board with dimensions of 100 x 160 mm (so-called Europe board). The main advantage of this pcb format is that the controller fits into 19" racks. Furthermore, many boxes of these dimensions are available, the pcb of the TNC2 having the same format, for example.
Despite of using highly integrated components integration could only have been realized by solely applying SMD components. However, assembling of the kit should be not only for experts but for hams in the first place, the design was adapted to the room available on the board. Single component groups could be simplified considerably due to exchange of the NEC V50 processor against the software compatible NEC V25. Following the processor exchange diverse digital control logic as well as I/O ports could be saved. Additionally, the new processor provides a second V24 interface and supports multitasking.
At the same time dynamic RAMS were substituted by static ones due to considerably lowering of price and dimensions of static RAMs. Now there was room enough that beneath the real-time clock status LEDs as well as a serial EEPROM could be integrated on the board. Deliberatly, no modem was included in the circuit design because evolution goes on very fast on this sector [2]. All signal lines are attached to a 96 pin indirect connector. The signal lines are arranged in a way allowing direct connection of the SCSI bus via back plane. The modem signals can be lead intersectionfree onto 20 pin flat cable connectors [2]. Additionally, two 8 bit I/O ports for control and measuring purpose are available at the indirect connector.
Because of the exorbitant high component density a two layer pcb could not be used. According to this and to improve EMC behavior, the pcb was carried out with six layers. Against common manner, the power planes were arranged on the outsides of the board to achieve maximum shielding of the routes. Thus, inevitable reciprocal interference between RF and digital components is minimized.

How does FALCon look like ?

After this glance backward on the evolution of FALCon a detailed description of construction and function of the whole design is given. On the pcb, following components are present:

- V25 CPU with a clock rate of 8 or 10 MHz,
- EPROM socket for EPROMs from 2764 to 274000 (8 kB to 512 kB),
- two static RAMs with 128 and 512 kB capacity, resp. (1 MB together),
- two or four SDLC/HDLC ports with DMA control,
- two V24 ports,
- SCSI controller,
- real-time clock,
- serial EEPROM,
- accumulator or buffer battery for real-time clock and RAMs,
- watchdog circuit,
- external power fail detect,
- twelve LEDs (three per HDLC/SDLC channnel),
- universal 8 bit I/O port,
- 8 bit input port, as well usable as AD converter,
- 96 pin indirect connector with all signal lines attached to.

The 8 MHz version of the the V25 can be used as well as the faster V25+ with 10 MHz clock rate. The advantage of V25+ processor is not only the higher clock rate but also the improved DMA block. The processor clock is derived from a quartz oscillator with double frequency. This clock rate is independent of the HDLC controller s working frequency which is generated seperately. Though, changing the processor frequency will not influence the HDLC baud rate. The processor frequency can be lowered software-controlled to reduce power consumption. In comparison with standard processors of the 80XXX series the V25 provides extremely improved abilities to support real-time multitasking. Among these features are for instance eight register banks which can be switched using a simple command. Changing the task, saving of the enviroment and restoring of registers is not necessary. These register banks can also be used for interrupt handling. At entry into the interrupt routine the pertinent bank is used and on exit the old status is restored automatically.
Additional features of the processor are: two freely programmable 16 bit timer, a time base generator, an interrupt controller, two DMA channels, an asynchronous serial interface equipped with a baud rate generator, an alternatively synchronous/asynchronous usable serial port with baud rate generator, three 8 bit I/O ports and an analog/digital 8 bit comparator port.
For support of the serial interfaces the V25 provides so-called macro service functions. These functions are equivalent to a micro program controlled DMA, so that the interrupt load is minimized when using the serial ports.

On the pcb, two of the three I/O ports are reserved for internal use and not available for the user. The third I/O port as well as the comparator port are attached to the indirect connector and though externally usable. The I/O port can be configured bitwise as in- or output with software commands. For instance, it can be used to interface the widely spread IIC bus. If required, the count of output lines can be multiplied by use of shift registers and latches. All I/O and comparator lines are equipped with pull-up and pull-down resistors, resp., to protect the CPU and avoid undefined states.

As mentioned before, the four SDLC/HDLC interfaces are realized with special components compatible to the 8530 providing an additional 4 channel DMA driver. During receiving and transmission no interrupts character by character are required. The transfer is handled only by the integrated high speed DMA. An interrupt is triggered only at begin and end of a frame to signal the CPU that action is required. Futhermore, the SDLC/HDLC controllers provide a 10 level deep state FIFO per channel which allows receiving of high speed back-to-back frames. For instance, while the CPU is reacting on the first frame received, data of the second or following frames are transferred to the main memory. Length and status of the first and following frames are held in the FIFO until the information had been processed by the CPU. The four ports can - as known from the SCC 8530 - be used as asynchronous interfaces if more than the two originally present asynchronous ports are needed. Another special feature of the SCC component used here is that all registers can be adressed without an index register. All modem signals are attached to the indirect connector (RTS, CTS, DCD, TxD, RxD, TxC, RxC, DTR and SYNC). Equal to the I/O ports, all input lines are equipped with pull-up and pull-down resistors, resp., to protect the CPU and avoid undefined states. Additional, there are three LEDs including drivers per SDLC/HDLC channel. These LEDs are controlled by the signals RTS (PTT), DTR (state) and DCD and though are equivalent to LEDs belonging to a TNC2 system.

The SCSI interface of FALCon is designed for use in high speed networks. Data transfer between the main memory and the SCSI bus is carried out using a channel of the internal DMA of the V25. The main part of functions required to control the SCSI bus is hard wired. Furthermore, the component contains all required drivers. If not needed, the SCSI component can be omitted.

Another features are the serial EEPROM, a watchdog circuit and an external power fail input.

The serial EEPROM uses jointly two lines of the 8 bit port. These two lines can be used for diverse tasks, because the select signal for the EEPROM is generated internally and independent of the two lines.

The watchdog circuit is equipped with the well-known component MAX 691. If an internal port line of the processor is not attended for longer than 1.6 seconds the system will be resetted.

The external power fail input can be used to watch an external power supply. With aid of this facility it is possible to check the unstabilized voltage before passing the voltage regulator. If this voltage descends under a chosen value, the power fail logic sends an non-maskable interrupt (NMI) to the V25 CPU. In this case, a software routine can run several

saving operations before the power supply of the CPU is breaking down. The power fail input has a trigger threshold of 1.25 V. If a higher voltage is to be watched, it has to be divided by resistors down to 1.25 V.

How to connect FALCon to the outer world ?

The answer on this question is dependent on the application. Minimum requirement is a 5 V power connection. A simple 5 V voltage regulator is sufficient for this purpose.
As there are no modems integrated to keep FALCon up to date, modems have to be connected externally, too. It is possible to use any common modem type. For users a adapter board is available which contains voltage regulators and connections for all interfaces. Both V 24 interfaces are attached to 9 pin DSUB connectors. In contrast to to the serial ports, the four modem interfaces are wired to a 20 pin high speed connector as described by Henning, DF9IC [2]. The SCSI interface can be externally connected to a 50 pin flat cable connector. Additionally, there is a litte interface for connection of a LCD display.
For use at net nodes, a back plane is in preparation which provides direct connection of several FALCons through the SCSI bus. On this pcb, there are also flat cable connectors for modems and V 24 interfaces. If needed, the SCSI bus can be active terminated on this board.

Up to now, a WA8DED compatible firmware with hostmode support via the V 24 port was implemented onto the new hardware as well as the newly developped software by Walter, DG9EP, called DigiWare. This program is compatible with the European FlexNet auto-router system, which uses a hop-to-hop procedure with link runtime measurement and other features. A node software compatible to NETROM or THENET has also been implemented.
At the time of this publication implementation of a hostmode interface via SCSI had already begun. With the aid of the SCSI controller and a small resident program on the host computer the system behaves similar to a common hostmode-TNC - with considerable advantages in speed and comfort for the user. This SCSI hostmode is especially suited to connect a mailbox system to a FALCon net node.

Interested users and programmers can order a user guide (available only in German). This is a complete documentation of hard- and software containing all technical data and the schematics.

Bibliography

[1] Werner Cornelius, DG3DBI: Schneller 16 Bit-Knotenrechner für Digipeater und Endbenutzer. - ADACOM Magazin Nr. 2, 1991.

[2] Wolf-Henning Rech, DF9IC: Modernes FSK-Modem - kompatibel zum Standard nach G3RUH. - ADACOM Magazin Nr. 2, 1991.

HEALTHNET VS. HF PACKET IN MOZAMBIQUE

Phil Gray, KA7TWQ/C9RPG
CARE Mozambique
660 First Ave.
New York, NY 10016

Abstract

Details of one HF Packet Radio network that failed after a four year attempt at implementation contrasted with one that was successful.

CONCLUSIONS

WHY HF PACKET MAY HAVE FAILED	WHY HEALTHNET MAY HAVE SUCCEEDED
The concept was unknown in Mozambique	The Ministry of Health was aware of both the concept and of Satel Life.
No Mozambican of high enough level was involved to assist with Telecommunications and customs problems.	We had three men at Assistant Ministerial level to help with TDM and customs difficulties.
My limited HF experience and how sensitive it was compared to VHF I was used to working.	An expert was on site to do the satellite station installation and integration.
My lack of understanding how one obtains money/purchases from CARE headquarters.	The equipment was a donation and free (providing we could import it without charge).
Not knowing the custom detail of importing electronics from Canada to the United States.	Having a man on CARE staff who understood local import regulations and procedures.
At least two separate and distant stations were needed to test but there was only <u>one</u> qualified person available to do so.	Only one station required.

CONCLUSIONS (CONTINUED)

HF	HEALTHNET
Antenna configuration of dipoles was not optimum for HF.	No problems arose with the omnidirectionals.
There was no money incentive.	There would be a huge savings in international communications.
Telephone land lines improved the last 18 months of the project so the BBS emerged as a usable substitute.	The BBS would augment and expand the station's reach up-country.
No locals confident enough with either radios or computers to want to learn the operation and system.	Several computer literate and interested doctors from the medical school.
Not enough free time to devote to the tests and training required.	Adequate free time plus a very personal interest.

HF PACKET RADIO PROJECT: MARCH 1987 -- APRIL 1991

I joined the CARE mission in Mozambique, Southern Africa in January, 1987. A nation-wide food distribution project was in process in that big, long country -- much of which was inaccessible due to the civil war in progress. CARE was the logistics branch of the government of Mozambique's Department for the Prevention of Natural Calamities (DPCCN). The telephone system in the nation had rapidly deteriorated since the 1975 revolution displaced the Portuguese. Communication up-country in early 1987 was accomplished 90% by HF radio, and that was limited almost entirely to the ten provincial capitals.

The year before in Ethiopia CARE and the Volunteers In Technical Assistance (VITA) had conducted fairly successful tests with HF Packet.[1] I asked for and studied those results. Three weeks into my work, it seemed there was a strong case for HF Packet. Thus began an exciting but frustrating four year campaign.

In September, 1987, I submitted a plan for a district radio network to connect 50 of Mozambique's 70 districts. In the proposal was the $30,000 for a Packet Radio net with one station to be set up for satellite work. I began to explore how to get the amateur frequencies from the Mozambique Telecommunications Department (TDM) for the satellite station and how to obtain the amateur license to work them. There were no known amateurs in country and the last license had been granted by the Portuguese government prior to 1974.

In February, 1988 I visited Gary Garriott at VITA HQ. We discussed the project and possible VITA assistance. My equipment in Maputo was not adequate to do an acceptable job of integrating packet stations for what HF requires, but VITA could do it. I returned to Maputo in late March and the Dutch letter of approval was on my desk: $259,709 -- $30,000 for packet! We were on our way.

On the 30th of May, I told VITA to proceed with the purchase and integration of ten packet stations. While that was going on, the National Director of the DPCCN was asked by my Country Director to assist with obtaining my amateur status. In addition, we held more meetings with TDM for frequencies for the district radios; little progress. Finally they were granted ten weeks later. About that time VITA also informed us they were still waiting for their money from CARE New York -- over two months waiting! So more telexes passed to and from VITA and to and from New York regarding payment. It was both embarrassing and frustrating. But I had to learn the CARE routines.

A bit later came another blow from Gary: the 7727C was no longer in production. Instead, the up-grade was an 8727 with similar specifications. (CODAN assured VITA the only things changed were improved electronics -- cable and connectors had not been altered.) We had no choice but to go with it. By this approval I didn't realize how it was to effect the duration, and hence, the ultimate end of the project. To start with, we learned all CODAN radios for Africa come from Australia. Any that go to the United States enter from Canada. As a consequence, the radio took over three months to get to VITA. And then testing was delayed because the microphone plug was non-standard and not available locally. This resulted in a further delay to make the required cables.

But on 24 APR 89, the equipment arrived. As I feared, however, the plugs and connections <u>had</u> been altered by CODAN. I began immediately to try to get the necessary changes made on

our 7727's, but by 04 JUL, we were still in trouble. Our frustration was enormous because this same equipment had worked in Ethiopia three years before. On the 12th, I faxed Gary a pin out schematic to see if he could see an answer I had missed.

I needed help and Johanessburg was closer than Washington. I called Peter Strauss -- a well known amateur in South Africa and a computer/communications expert. He arrived the end of September, 1989. After two days, suspecting frequency and antenna problems, he took one of our TNCs back to test. He returned the 23rd of October with his own station (and accompanying visa and customs headaches). Our highest frequency was adequate for some tests to his station in Joburg -- 800 KM away, but the link was not strong. We traveled to two up-country sites to try connecting with Maputo. This was a serious error: one of us should have stayed back to work the base station. The results were discouraging.

Peter's report arrived on 02 NOV and basically had two recommendations: 1. -- Change the antennae to beams, and 2. -- Get higher frequencies. Of the two, at least we had some control over antennas. Getting different frequencies from TDM had been nearly impossible and we couldn't hope to do it again.

In February, Peter and I had some tantalizing attempts connecting with a Beacon from his station. So he decided to try once more to get us on the air. I saw him in Joburg on my way out for home leave in March and we made plans for my return. He was enjoying his flying lessons.

In early May, we resumed HF tests between Maputo and Joburg. But in the next few weeks three things happen that took the heart out of the packet project and it never really recovered. First, and most tragic, Peter crashed and died during one of his flying lessons. Second, I got our International EMAIL system working and we became connected into the rest of the CARE missions in the world. Third, I installed the Remote Bulletin Board System (BBS) and started the process of connecting with the ten provinces by land line. On 27 APR 91 I sent an EMAIL to VITA thanking Gary for all his assistance and good wishes, but it was the end of the Packet project for Mozambique.

Note: After all the problems I had with the 7727, in early 1992, CODAN sensed there was a market and began to sell their 8525-B SSB transceiver fitted out with a CODAN MODEM and a LapTop computer for a "Data over HF System."

SATELLIFE (HEALTHNET) PROJECT: DECEMBER 1987 -- JULY 1992

I first came to know of the SatelLife Project when I read about it in the 02 NOV 87 issue of Amateur Satellite Report (ASR)[2]. By December, I was on their mailing list as a volunteer willing to participate in their activities however I could.

From May, 1988, until June 1990, I heard nothing. Then in the June, 1990 issue of 73 Amateur Radio[3] appeared a report describing SatelLife and naming Dr. Charles Clements as the new Executive Director. I wrote him on 15 JUL 90, to reintroduce myself and my location. He wrote back in December to say things were moving very fast and several agencies have gotten involved with them including VITA and IDRC. Mozambique was ideally suited to be part of the project if we wished. In fact, SatelLife's field director in Africa, Mr. Mackie McCleod, had even met with the Minister of Health in Maputo. I replied in late February, 1991, saying I would pursue the required permission for frequencies and having the Minister of Health on our side would be most beneficial.

On 07 NOV 91, Dr. Clements sent an introductory letter to Dr. Jorge Barreto of the National Institute of Health (NIH) in Maputo. He spoke about SatelLife, IDRC, engineer Edson Pereira (who was installing ground stations in the region) and offered to donate a satellite station to the NIH.

I went immediately to meet Dr. Barreto. We devised a strategy to obtain the required licensing and frequencies. We also needed to get the donated equipment into the country as cheaply as we could -- free if possible! On 21 NOV, I informed SatelLife what was transpiring.

Dr. Clements said instructions for shipping the hardware had to be sent to Boston immediately. On the 26th of November, we held another strategy meeting. This time we involved the Chairman of the University Medical School and the Director of the General Hospital. It decided to have the equipment addressed to the hospital -- which would allow it to enter duty free. Dr. Barreto was asked to use his influence and connections downtown to obtain licensing on the condition the station was located at the medical school.

While we now knew where to send them, we were not sure <u>what</u> documents we needed to insure duty free importation. For this

we got help from my CARE. The Mozambican in charge of procurement told me what kind of documents we should have available to present to customs officials at the airport. I requested these from SatelLife on 09 DEC. They were immediately sent to us by FAX. On the 11th the cargo arrived at the airport and the usual customs hassles began. More on that later.

On the 21st Edson arrived. I invited him to dinner and asked Don Findley if he would like to join us. Don and I had been planning some VHF packet experiments. He was most interested in being a part of this satellite venture. So Monday morning the three of us met first with Dr. Barreto to introduce Edson and explain what we would be doing for the next few days. Then all of us went to the Medical School to meet with the Chairman of the Faculty and the Director of the General Hospital. We all noted the equipment had not cleared the airport yet so the Director sent his men to customs to find out why. Then we did a tour of the building to determine the best location for the station. Dr. Barreto and Edson needed to meet for a while so Don and I took this opportunity to go visit TDM and check on frequencies and my amateur license. For four years I had not felt it so necessary to have my license that I needed to do anything out of the ordinary to obtain it -- a few informal lunches, the occasional coffee, etc. But we had to have that document now. I carried with me a bottle of expensive whisky. In ten minutes I walked out with my new amateur license: C9RPG.

After lunch we began to worry about customs. Still no equipment was cleared. Customs bureaucrats everywhere in the world simply hate duty free imports, and the personnel here were no different. They had stalled for nearly two weeks. It was time to bring our the heavy artillery: Dr. Barreto was chosen to go to customs the next morning. He left at 08:30. Hoping to help by just being there, Don, Edson and I got there at 10:00. We walked in and just stood behind Dr. Barreto who was seated arguing with some official. It seemed to work. A few minutes later Edson left for the airport later drove up to the Med-school with all the boxes.

Edson's time with us had been reduced from six days to little over three. We began immediately and continued through Christmas day. At 15:30 on December 26th, we sent our first message: to SatelLife informing we were up and working on UOSAT 14! Edson left the next day for South Africa where the head of South Africa AMSAT, Mr. Hans Van de Groenendaal received him.

In February of this year, all amateurs and SatelLife stations were traumatized when the great switch between UOSAT3 and UOSAT5 took place. But since then, we have mostly been waiting to use the SatelLife station as it was designed: exchange medical information among users. Also planned is a land line BBS that will connect the Maputo station to the provincial capitals. CARE operates one such BBS now. We expect to be operational by year's end. We have formed an Users Council as required by all SatelLife stations to make the most democratic use of both the satellite station and the BBS and look forward to an exciting future.

* Paper for the 11th ARRL Amateur Radio Computer Networking Conference at Fairleigh Dickinson University; 07 NOV 92. The opinions expressed herein are entirely the author's and not necessarily those of CARE International, VITA or SatelLife.

References

[1] R. A. Whiting & H. P. Snyder, Report and Recommendation Packet Radio Feasibility Demonstration Project 6 June 1986. (Washington DC: VITA, 1986).

[2] Moscow Space Forum Highlights Future Space Activities, Amateur Satellite Report (November 2, 1987; No. 161): page 1.

[3] SatelLife Packet, 73 Amateur Radio, (June 1990): page 7.

[4] PACSAT Proliferation, OSCAR Satellite Report, (September 9, 1991; No. 228): page 2.

SOME RECENT AMATEUR USE OF FEDERAL STANDARD AUTOMATIC LINK ESTABLISHMENT (ALE) SIGNALING

Bob Levreault (W1IMM) and Ken Wickwire (KB1JY)

INTRODUCTION

This paper describes some recent experience on the HF bands with equipment operating according to the new federal standard for automatic link establishment (ALE). Some data collected on authorized frequencies outside the ham bands (where interference is less bothersome) are also presented to illustrate some of the analysis possibilities offered by ALE systems.

In our experiments on the amateur voice and digital sub-bands we have established links using Federal Standard 1045 ALE controllers and then used the links for voice or data exchanges. The controllers contain modems, and software that implements the ALE linking protocol and other functions connected with network operation based on ALE. Details on how these ALE controllers work have been given recently in several articles in *QEX* and *QST*.

The experiments, which began in June, have been conducted mainly to see how well ALE works in the noise and interference conditions of the amateur bands. We have interpreted ALE in our experiments as a means for establishing links that will be used for conventional ham voice or data traffic, and have tried to keep our use of the 375-bps, 8-ary FSK ALE waveform brief and at relatively low power (100 watts output).

Most of the links have been between Boston (KB1JY and W1IMM) and Raleigh, N. C. (NT4T), with a few between Boston and Cedar Rapids, Iowa (WA0IQM). The Boston-Raleigh link is about 500 miles long and the Boston-Cedar Rapids link about 1000 miles long. Our antennas are broad band or tuned wires (in Boston and Cedar Rapids) and a tuned whip (in Raleigh).

These experiments may have been the first use by hams of ALE in the ham bands, although hams (and many others) have been experimenting with the federal standard technique in other parts of the HF spectrum for about three years.

We have not run the ham experiments on a regularly scheduled basis, so this report gives only an indication of how well ALE works and how its performance compares with that of the conventional modes of digital signaling in the ham bands (Morse code, HF packet, AMTOR or ASCII). When permission is granted for regular use of ALE in the amateur bands, our approach should be replaced by systematic data collection.

Here's a summary of how ALE works:

The ALE controller uses digital signal processing to automatically

- sound channels (in one- or two-way modes),
- collect and store data on channel quality,
- exchange channel quality data with other stations,

- chose the best channel (frequency) for communications,
- call a station or stations, and set up a link on the chosen frequency using the ALE protocol,
- alert operators of the established link for subsequent transmission of data or voice traffic, and
- exchange short, stored digital messages if desired.

The linking exchange is three-way: *call-response-confirmation*, and all three legs must be successful for link establishment. The short messages appear on front-panel displays of the controllers and are called automatic message display (AMD) messages.

Both ALE data frames (containing station addresses and channel quality measurements) and those of the short messages allowed by the standard are protected against bit errors by means of a powerful combination of (23,12)-Golay coding, interleaving and three-fold diversity (bit repetition) at the transmitting modem. The receiving station carries out the corresponding Golay decoding, de-interleaving and majority-vote decoding. The Golay code provides protection mainly against isolated bit errors caused by static, etc. The interleaving and repetition protect mainly against "burst errors" caused by the inter-symbol interference associated with HF multipath and by unwanted radio signals.

Data traffic can be sent over ALE links by Morse code, unprotected FSK (ASCII, BAUDOT), binary (A)FSK with some error control (HF packet, AMTOR, PACTOR), by the 8-ary FSK plus error control of the ALE waveform itself, or by some other efficient waveform with error control (for example, CLOVER, or one of the new MIL-STD-110A waveforms).

THE AMATEUR EXPERIMENTS

These experiments took place on an agreed-upon and stored set of 6 frequencies in the 75-, 40-, 30-, 20-, 17- and 15-meter bands.

The ham-band experiments started with either sounding or a link quality analysis (LQA) exchange. Sounding involves a set of one-way transmissions on a stored set of frequencies that allow stations scanning the set to measure channel quality. An LQA exchange is a similar *two-way* exchange of channel quality data. In each case, a sound or LQA attempt will generally result in data collection on only a subset of the stored frequency set; namely, those frequencies that propagated well enough to allow the corresponding stations' addresses to be read by the receiving stations.

Table 1 gives an excerpt from our ALE data log for ham-band experiments run between Boston and Raleigh on 26 August 1992 at about noon EDT. It shows that an LQA initiated by Boston got responses on 14 and 7 MHz. Channel quality in each case was high enough for communication with the ALE waveform (an AMD with no errors) and with ASCII (where errors were usually noted). Some details on the format of the log output are given below.

After finding out what frequencies were good, we then choose one of them manually for linking, or let the modem pick what it thought was the best one, and then try to set up a link automatically. In experiments carried out around noon between Boston and Raleigh, the chosen frequencies were usually 10 or 14 MHz, which agrees with IONCAP predictions of about 14 MHz for the average MUF for Boston-Raleigh in summer. In experiments run in the evening,

the preferred Boston-Raleigh frequencies were 7 or 10 MHz (the IONCAP MUF was about 11 MHz). The preferred frequencies for Boston-Cedar Rapids were a few megahertz higher.

Table 1. Data Log for Ham-band Experiments on 26 August 1992

NT4T/IMM ALE/ASCII 26.8.92											
LQA de IMM with longwire											
Date						Freq.	SNR		BER		Comb.
8/26/92	GMT	Event		From	Ch		To	Fr	To	Fr	Score
8/26/92	16:57:36	Lqa_initiated_to	NTTP	IMM	35						
8/26/92	16:57:57	Lqa_initiated_to	NTTP	IMM	34						
8/26/92	16:58:14	Lqa_initiated_to	NTTP	IMM	33	14.08	*	20	0.021	0	99
8/26/92	16:58:34	Lqa_initiated_to	NTTP	IMM	32						
8/26/92	16:58:52	Lqa_initiated_to	NTTP	IMM	31	7.093	*	19	0.083	0	81
8/26/92	16:59:12	Lqa_initiated_to	NTTP	IMM	30						
* No measurement received from NTTP											

In 70 or 80 trials, automatic link establishment almost always worked on the first try[1]. On each established link we initially send an Automatic Message Display (AMD) message about 80 characters long. These messages appear on the front panel of the receiving controller and are protected by ALE error control. We send AMDs for comparison with digital signaling using various ham waveforms (HF packet, ASCII, AMTOR or Morse code).

The AMD messages always arrived without error over the channels chosen by the modem, as has automatically sent Morse code, which we demodulate by ear.

Voice communications were usually readable, but were carried out against a background of strong summer static, caused probably by lightning discharges a long distance from our stations.

Packet messages, whose errors are controlled by a form of automatic repeat request (ARQ), suffer sometimes from the well known effects of multipath fading: packets about 80 characters long occasionally took two or three tries to arrive correctly. Our KAM or PK-232[2] packet modems' tuning indicators suggested that multipath (rather than noise) was the cause of this.

ASCII transmissions are of course unprotected by any error control. We sent 80-character ASCII messages over the same frequencies chosen by the controller for ALE. We judged reception quality by counting character errors. Character error rates between 5 and 20% were common for ASCII, and most errors came in bursts.

Similar things happened with AMTOR FEC (Mode B), although the error rates were a bit lower than for ASCII. This is expected since AMTOR FEC uses two-fold character repetition and a CRC error-detecting code for error control. Error rates were not reciprocal, a reflection, perhaps, of the different antennas, or different noise levels, or both.

---

[1] On one notable occasion, however, there was so much noise and interference on 7 MHz that an attempt to link at about 2200 GMT took 4 tries. AMTOR, on the other hand, could not be read at all.

[2] It is instructive to note that because of the fact that ALE takes place on *fixed* channels, it is not possible (or at least not easy) to communicate on an ALE channel with amateur multi-mode controllers that use different FSK mark and space frequencies. This is the case with the KAM and PK-232. Fortunately, we were able to change the KAM's default mark and space to the mark and space used by the PK-232.

The sounding and LQA mechanisms of the ALE system make it easy to collect data on short-term or long-term channel quality (useful in network analysis) and on antenna performance. The controller can be programmed to sound (a form of broadcasting to any stations scanning the frequency set) on schedule or to call and exchange channel data with a particular station. Either method allows receiving stations with data storage capabilities (a PC with a hard disk) to collect channel quality data systematically. Current equipment measures channel quality in terms of signal-to-noise ratio (SNR) and bit error rate (BER), which are measured independently.

EXPERIMENTS OUTSIDE THE HAM BANDS

Channel Quality Measurement

As an example of channel quality data collected with the ALE protocol consider the spreadsheet excerpt shown in Table 2, which was made using data collected automatically from an RS-232 port on the ALE controller in Boston. The excerpt refers to two sets of linking exchanges made by "MIB" (Boston) with "MTR" (in Virginia, about 400 miles) and "SUN" (in Florida, about 1000 miles) on various frequencies.

In each case, the SNR (in dB, maximum value = 30) and bit error rate (BER) on the links were measured in Boston. These are listed in the **From** columns. In the case of the SUN links, Boston also recorded the SNR and BER that *SUN* measured: these are in the **To** columns, and were sent to Boston as part of the LQA exchange that normally occurs during the three-way linking process. In the case of MTR in Virginia, only the BERs are two-way: equipment incompatibilities prevented a two-way SNR exchange. The **Comb Score** column contains scores (maximum value = 120) that reflect the overall quality of each link; these were calculated by the Boston controller from the two-way SNR and BER measurements.

Figures 1 and 2 show the variation of two-way SNRs and BERs on the links with SUN at 10.4 MHz listed at the bottom of the spreadsheet in Table 2. These links were made by repeatedly commanding the Boston controller to try 10.4 MHz, and the measurements were taken over the course of about 5 minutes. "Measured at MIB" refers to the measurements made in Boston of the quality of the SUN signal (in the **From** column), and "Measured at SUN" refers to the measurements made in Florida of MIB's signal and sent as ALE orderwire data back to Boston (in the **To** column). Since the controllers measure SNR and BER by independent means, the BER can't be derived precisely from the SNR and vice versa.

It can be seen that SUN's SNR was significantly greater than MIB's during all of this period, perhaps a result of different background noise levels at each end of the link. Both signals suffered somewhat more variation during the first 2 minutes than the last 2, with the SUN signal changing significantly faster during the first 2 minutes than the MIB signal. The early variations probably reflect the presence of radio interference; the interference was probably not present during the last 2 minutes.

It is interesting to compare this measured performance with that predicted by the IONCAP program used by *QST* and many others in forecasts of DX operation. IONCAP predicts that this link has an optimal working frequency (FOT) of about 14 MHz at 2200 GMT, and a reliability (probability that the SNR required for ALE will be achieved) of 70%. The reliability at 10 MHz is about 46%. (In this prediction we assumed a sunspot number of 100, equal noise levels at each end of the link, and the use of zero-gain isotropic antennas.) Note that the use of 15 MHz between 21:55 and 21:56 resulted in a BER of zero for both ends of the link. This suggests that there was less multipath (and inter-symbol interference) at 15 MHz than at 10 MHz (the predicted MUF was 18 MHz).

The markedly different SNRs recorded at each station (assuming that the controllers are using comparable measurement techniques) suggest that the noise levels or antenna gains are in fact different. It should be kept in mind that IONCAP predicts *monthly averages* of the MUF, FOT and SNR, and says nothing about radio interference, so differences between its predictions and actual performance at a particular time are to be expected. IONCAP should be viewed in the context of ALE systems as a means for choosing the set of frequencies to be tried by the controller; we should *not* expect it to tell us the frequency that will actually be chosen.

Table 2. Data Log for MIB/MTR and MIB/SUN Links on 6 May 1992											
Date	GMT	Event	To	Frm	Freq		SNR		BER		Comb
Individual calls							To	Fr	To	From	Score
5/6/92	21:39:45	Linked	MTR	MIB	9.973	USB	- -	10	0	0.029	80
5/6/92	21:40:22	Linked	MTR	MIB	7.422	LSB	- -	12	0	0.007	86
5/6/92	21:40:50	Linked	MTR	MIB	7.422	LSB	- -	18	0.014	0	97
5/6/92	21:41:20	Linked	MTR	MIB	7.422	LSB	- -	16	0.021	0.021	89
5/6/92	21:41:44	Linked	MTR	MIB	7.422	LSB	- -	13	0.014	0	86
5/6/92	21:42:18	Linked	MTR	MIB	7.422	LSB	- -	11	0.014	0.051	78
5/6/92	21:42:41	Linked	MTR	MIB	9.973	USB	- -	30	0	0	120
5/6/92	21:43:04	Linked	MTR	MIB	9.973	USB	- -	18	0	0	100
5/6/92	21:43:29	Linked	MTR	MIB	9.973	USB	- -	10	0	0.051	77
5/6/92	21:43:51	Linked	MTR	MIB	7.422	LSB	- -	18	0.014	0.029	94
5/6/92	21:44:16	Linked	MTR	MIB	7.422	LSB	- -	18	0.021	0.014	94
5/6/92	21:44:39	Linked	MTR	MIB	7.422	LSB	- -	16	0.007	0	95
5/6/92	21:45:01	Linked	MTR	MIB	7.422	LSB	- -	18	0.021	0	95
5/6/92	21:45:24	Linked	MTR	MIB	7.422	LSB	- -	10	0.029	0.044	74
5/6/92	21:45:46	Linked	MTR	MIB	9.973	USB	- -	21	0	0	104
5/6/92	21:46:12	Linked	MTR	MIB	9.973	USB	- -	30	0	0	120
5/6/92	21:46:35	Linked	MTR	MIB	9.973	USB	- -	30	0	0	120
Individual calls											
5/6/92	21:55:01	Linked	SUN	MIB	15.71	USB	7	10	0	0	55
5/6/92	21:55:25	Linked	SUN	MIB	15.71	USB	11	15	0	0	69
5/6/92	21:55:47	Linked	SUN	MIB	15.71	USB	11	30	0	0	84
5/6/92	21:56:15	Linked	SUN	MIB	15.71	USB	10	15	0	0	66
5/6/92	21:57:03	Linked	SUN	MIB	10.42	USB	6	19	0.021	0	63
5/6/92	21:57:25	Linked	SUN	MIB	10.42	USB	8	17	0.036	0	63
5/6/92	21:57:50	Linked	SUN	MIB	10.42	USB	8	30	0.007	0	77
5/6/92	21:58:11	Linked	SUN	MIB	10.42	USB	8	14	0.029	0.007	59
5/6/92	21:58:32	Linked	SUN	MIB	10.42	USB	5	21	0.051	0	59
5/6/92	21:58:54	Linked	SUN	MIB	10.42	USB	8	30	0	0	79
5/6/92	21:59:16	Linked	SUN	MIB	10.42	USB	6	13	0	0.007	58
5/6/92	21:59:38	Linked	SUN	MIB	10.42	USB	9	30	0.014	0	78
5/6/92	21:59:59	Linked	SUN	MIB	10.42	USB	8	30	0.029	0	75
5/6/92	22:00:22	Linked	SUN	MIB	10.42	USB	10	30	0	0	83
5/6/92	22:00:43	Linked	SUN	MIB	10.42	USB	11	30	0	0	84
5/6/92	22:01:05	Linked	SUN	MIB	10.42	USB	11	30	0.007	0	85
5/6/92	22:01:27	Linked	SUN	MIB	10.42	USB	11	30	0	0	84
5/6/92	22:01:47	Linked	SUN	MIB	10.42	USB	10	30	0.007	0	81

Figure 1. Measured SNRs @ 10.4 MHz

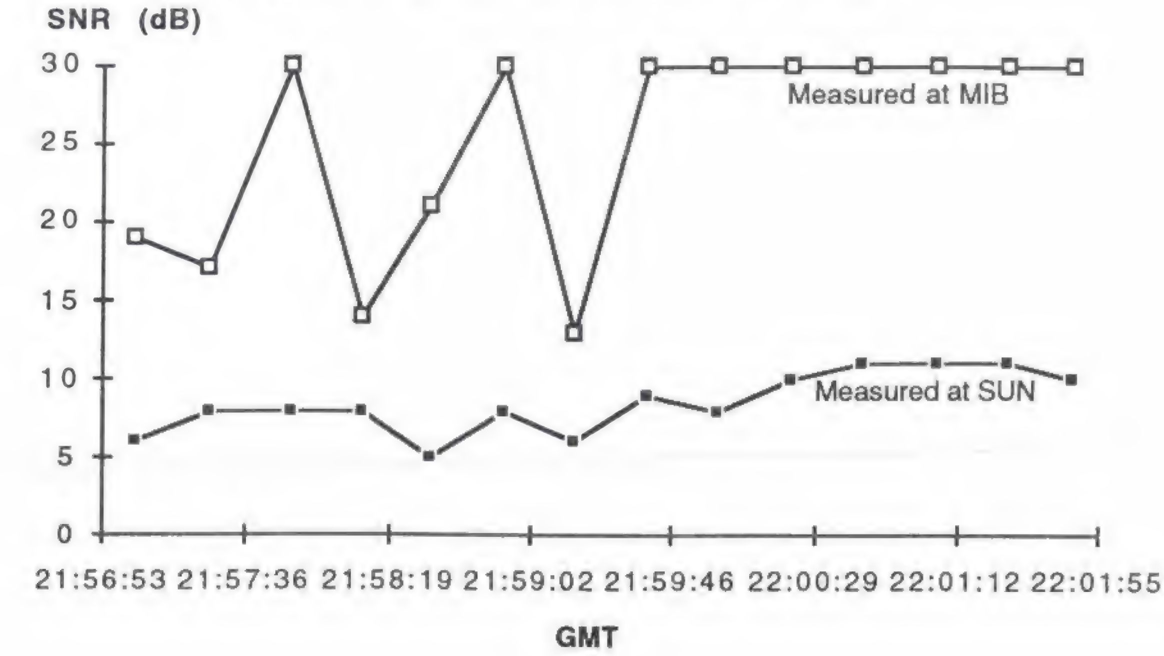
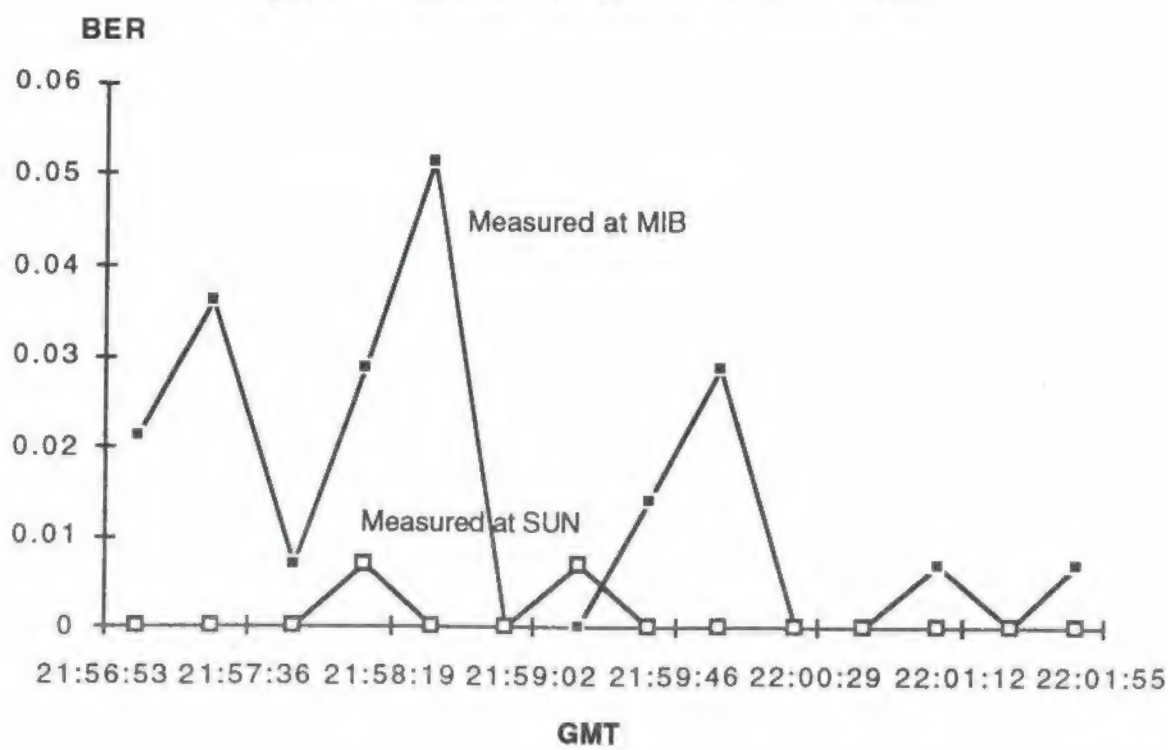


Figure 2. Measured BERs @ 10.4 MHz



This example is typical of what is often observed on such a link, and it points to the importance for effective HF digital signaling of an automatic means for measuring channel quality and establishing links. In this case, SUN's controller (using the Combined Scores) may well have chosen 10.4 MHz for a link attempt with MIB, but the MIB controller would probably have looked for a channel with higher Combined Score.

Comparison of Antenna Performance

One way to compare antenna performance is to carry out an LQA exchange on a set of frequencies with each antenna at nearly the same time. As an example of this, consider the data log excerpt shown in Table 3. It applies to a pair of LQA exchanges initiated by MIB (near Boston) with MOT (near Chicago, Ill). For the first exchange, MIB used a 100-ft, broad band, omni-directional dipole, and in the second (about 2 minutes later) a 250-ft, resistively terminated, sloping long wire pointing south. (The dashes in the **To** columns at 7.4 MHZ indicate missing measurements.)

Table 3. Dipole vs Long Wire on 20 February 1992												
100-ft Omni-directional Dipole								SNR		BER		Comb
Date	GMT	Event	To	Frm	Ch	Freq		To	Fr	To	Fr	Score
2/20/92	22:51:52	Lqa_initiated_	MOT	MIB	10							
2/20/92	22:52:27	Lqa_initiated_	MOT	MIB	9							
2/20/92	22:53:03	Lqa_initiated_	MOT	MIB	8							
2/20/92	22:53:39	Lqa_initiated_	MOT	MIB	7							
2/20/92	22:54:14	Lqa_initiated_	MOT	MIB	6							
2/20/92	22:54:30	Lqa_initiated_	MOT	MIB	5	10.42	USB	24	30	0	0	111
2/20/92	22:54:46	Lqa_initiated_	MOT	MIB	4	9.973	USB	20	30	0	0	104
2/20/92	22:55:21	Lqa_initiated_	MOT	MIB	3							
2/20/92	22:55:57	Lqa_initiated_	MOT	MIB	2							
2/20/92	22:56:33	Lqa_initiated_	MOT	MIB	1							
250-ft Long Wire with resistive termination												
2/20/92	22:58:02	Lqa_initiated_	MOT	MIB	10							
2/20/92	22:58:37	Lqa_initiated_	MOT	MIB	9							
2/20/92	22:59:13	Lqa_initiated_	MOT	MIB	8							
2/20/92	22:59:49	Lqa_initiated_	MOT	MIB	7							
2/20/92	23:00:24	Lqa_initiated_	MOT	MIB	6							
2/20/92	23:00:40	Lqa_initiated_	MOT	MIB	5	10.42	USB	16	19	0	0	82
2/20/92	23:00:56	Lqa_initiated_	MOT	MIB	4	9.973	USB	18	30	0	0	98
2/20/92	23:01:11	Lqa_initiated_	MOT	MIB	3	7.422	LSB	- -	13	- .	0	23
2/20/92	23:01:46	Lqa_initiated_	MOT	MIB	2							
2/20/92	23:02:22	Lqa_initiated_	MOT	MIB	1							

The LQA scores confirm what we might expect: the omni-directional antenna does better on 9.97 and 10.42 MHz than the long wire, whose main lobes are probably not pointed toward Chicago. At 7.4 MHz, however, the long wire is apparently the winner. Of course, there are a number of possible explanations for this (for example, a 7.4 MHz lobe that *is* favorable for the east-west path, or a sudden burst of noise on 7.4 MHz when LQA was tried with the dipole), so no firm conclusion can be reached on the basis of a single observation. Nevertheless, the LQA

logging feature of some ALE systems is a very useful aid to the choice of antennas and antenna siting when coupled with a systematic measurement plan.

SOME ISSUES ARISING IN ALE USE ON THE HAM BANDS

Among the ALE issues that will have to be discussed and resolved by amateurs in the near future are:

- deciding what frequencies (or bands of frequencies) should be allocated for use by amateur ALE stations,

- working out effective protocols, waveforms and interfaces for sending data over links established and maintained with ALE (AX.25 packets encapsulated in ALE frames and protected by ALE error control, packets sent with the CLOVER waveform, or with other waveforms, such as those of MIL-STD-110A or an international standard? Data interface via the radio's audio port or an RS-232 data port?),

- coordinating the frequencies and callsigns used in network operation, and

- setting up an orderly and *standard* system for gathering, displaying and analyzing data on ALE performance in the ham bands.

More than 2000 federal standard ALE controllers are now used worldwide in commercial and military short-wave communications, and it is only a matter of time before a version becomes available for legitimate (if restricted) use in the ham bands. We hope that our experiments will increase interest among hams in this new and exciting technology.

ACKNOWLEDGMENTS

We are grateful to our colleague Ron Dugay for regular assistance with maintaining the equipment and collection of data. Bill Beamish of the Harris Corp. was kind enough to lend us the ALE controller we use in Boston. Bill Jackson (NT4T) of the Mackay Corp. in Raleigh N. C., and Gene Teggatz (WA0IQM) of Rockwell in Cedar Rapids, Iowa, joined us in the experiments.

Questions or comments to Ken Wickwire, 232 North Road, Apt. 17, Bedford, Mass. 01730.

The ROSE X.25 Packet Switch Application: "CWID"

Thomas A. Moulton, W2VY

The Radio Amateur Telecommunications Society

ABSTRACT

The purpose of this paper is introduce and describe the CWID application for the ROSE X.25 Packet Switch. In certain countries the need to support this function has been raised as a regulatory issue. Now, at the option of the System Manager, the CWID application may be loaded and conform to this requirement.

INTRODUCTION

A general design principle for ROSE X.25 Packet Switch Applications is to only consume memory resources for features that are needed in a particular Switch or Network of Switches. Any of the optional applications may be uploaded over the network on an "as-needed" basis.

Some of applications that were developed required installation of special hooks in the ROSE X.25 Packet Switch EPROM code, but generally these hooks only use 20-30 bytes of code space.

Some of the applications that currently exist are LOADER, CONFIG, USERS, HEARD, INFO and other special applications created to debug the system.

These applications appear in the network as network reachable destination end points. For example: "C INFO V W2VY-3,201555", would provide local network information for the 201 Area Code.

An application should not disrupt the normal operation of a Switch, nor should it introduce excessive delays or use large amounts of memory. It also may be deleted when no longer need by the network users or managers, thereby releasing memory for use as buffer or application space.

EXISTING CWID METHODS

The existing user TNC code supports CWID by keying the PTT and then toggling some lines to manually send the CW. This method could not be used within the ROSE X.25 Packet Switch because many hooks would be needed, and the timers required for the proper keying rate did not exist. This approach would clearly consume more code space than could be justified for an optional application.

NEW METHODS

One method would be to fill a packet with data that would be sent at 1200 baud in an attempt to sound like CW. There would be two data patterns, one for "ON" and one for "OFF", but the packet required to send "DE W2VY" would be 722 bytes long and more than 1000 bytes for longer callsigns. It is not clear how well this would sound nor if it could be identified as an official CWID. The packet size would be quite large and a large amount of CPU time would be spent generating the packet.

It was desired that the CWID sound much like existing systems. The I/O Chip used in most TNCs is either the Z8440 SIO

or the Z8530 SCC. These chips have a lot in common and with some examination of speeds (WPM) and bit rates some interesting things are possible. CWID should run at about 20 WPM, which comes to 16 BPS (WPM / 1.25 = BPS). The SIO is generally configured for a X1 transmit clock, which means that the transmit clock is running 1200 BPS (1500 WPM). With a single I/O Write the clocking rate (divisor) can be changed to X16, X32 or X64. The result of the change is that the clock would be divided to generate the true bit rate. The rate of X64 yields 16.75 BPS or 23 WPM. At this rate the longest callsign would require 13 bytes of data.

The hooks required to implement the CWID in the ROSE X.25 Packet Switch EPROM is simply code to set and reset the baud rate divisor. The "CWID" packet can then be included in the normal data stream without major disruption to the port.

CODE SELECTION

The next step is to perform tests to pick the binary sequences to represent a DIT, DAH, Letter Space and Word Space. The one sequence that must be avoided is 5 1's in a row, because the HDLC encoding will insert a 0 bit for its transparency coding.

The sequence sent for a DIT is 01, DAH is 0001, Letter Space is 11 and Word space is 111. The word space should be longer, but we would start to get transparency inserted zero's, which made readability difficult.

IMPLEMENTATION

When implementing the ASCII - Binary CW coder a few interesting items came up. First, in data communications the Low Order bit is always sent first. The meant that all the data patterns had to be sent through a software shift register to determine the true byte pattern that needed to be sent.

Due to the nature of the SIO, there will be a SDLC Flag or two at the start of the transmission and a CRC or ABORT at the end of the frame. To improve readability the end of the CWID text is indicated by an AR.

HIDDEN APPLICATION

The CWID Application is not a normal application. Since users don't interact with it over the air, It can be bundled into another application. The CWID has been included in a version of the INFO application, file name INFOCWID.LOD.

The INFO Application contains a couple of features, an INFO destination that will return configurable informational text, and plain-language connection status messages to supplement the terse codes provided in the ROSE X.25 Packet Switch EPROM code. In addition, the text can be in either English, Spanish, or German.

CONCLUSION

This exact scheme will only work for 1200 baud packet using Bell 202 modems, but is an example of how other signalling systems can be built into existing systems.

The ability to remotely load applications into network switches is the single most important platform feature of the ROSE X.25 Packet Switch and allowed for the rapid implementation and testing of this valuable new feature.

The ROSE X.25 Packet Network MS-DOS Device Driver

Thomas A. Moulton, W2VY
J. Gordon Beattie, Jr., N2DSY

The Radio Amateur Telecommunications Society

INTRODUCTION

There has always existed a great barrier to the casual programmer (a.k.a. hacker) who wants to write a program that communicates with another system. The process of buffering and sending and receiving characters is tedious and wards off all but the very committed programmers.

The ROSE X.25 Packet Network MS-DOS Device Driver is going to redefine the rules as far as how hard it is to establish and maintain a connection through a packet network.

EXISTING METHODS

Presently there are a few methods a programmer can use to access the packet network. They are:

- INT 14 to a TNC

- INT 14 to a Virtual TNC

- INT 14 to a KISS TNC

- INT 14 to a TNC in Host Mode

Each of these have their own advantages and disadvantages, but they have one painful thing in common, INT 14, which is a simple character by character transmission and reception method.

There are also Terminate-and-Stay-Resident (TSR) programs that expand the functionality of INT 14, but there are also problems of incompatibility with other terminal programs.

The biggest problem is that there is no official method to access the COM ports from most programming languages. They do however provide methods for accessing "Files". What is needed is a scheme that lets the network appear to the user and programmer as a File.

MS-DOS DEVICES

It is a very straight forward process to set up a MS-DOS Device Driver to control a COM Port. There are two types of MS-DOS Devices, Block Devices and Character Devices. Both can be accessed as a File and read/written with standard high level language statements.

A Block Device has a fixed record length and must be formatted like a disk drive. This will not suit our needs.

A Character Device is a serial stream of information, can be named with up to an 8 character name, and MS-DOS does not place any constraints on the format of the information.

DEVICE ROSE:

To keep the interface as simple as possible all of the connection information should be described in the file name.

In a ROSE Network the user needs to specify the Callsign and Network Address of the destination station. For example the Command "C W1AW V W2VY-3,203666" will establish a connection to a station with the callsign W1AW at the Switch that provides coverage to Network Address 201666. To establish the same connection using the ROSE Device Driver, the user would open a file with the name "ROSE:W1AW@203666". The user then may read and write the file to interact with the network. [Note: The terms "Read" and "Write" are not being used in the context of an editor reading in a file, but to indicate an interactive reading and writing of the connection to the remote user.]

In MS-DOS a program can have a large number of files open at a given time. The ROSE Device Driver also allows for this, in fact since it is a true Device Driver, it will automatically allow open files from different programs running under a multitasking environment.

EXISTING SOFTWARE

In order to be compatible with the existing software that has already been written for packet radio, the driver will also support the MBBIOS/COMBIOS interface via a simple Virtual TNC much like what was developed in the G8BPQ Switch. This interface will be provided to allow a simple method for users to use the Driver with existing software until the developers can modify their code to the new, simpler interface. This virtual TNC will also support KISS in later releases.

IMPLEMENTATION

The Driver will communicate to the outside world through a COM port to a ROSE X.25 Packet Switch asynchronous port or matrix of Switches using the Asynchronous Framing Technique (AFT).

Other configurations that will be supported include COM port to KISS TNC, as well as interfacing to the HDLC cards currently in use.

The code is being developed in C and is still underway. There are a lot of hooks in MS-DOS that needed to be trapped. The File Open, Close, Read, and Write interface is complete as well as the Timer Tick and Multitasking Interrupt. As of the writing of this paper, the COM port interrupt handler and the glue between the ROSE code and the Driver code need to be written.

CONCLUSION

The ROSE X.25 Packet Switch-based networks have been growing rapidly over the last two years due to their simplicity and rich functionality. With this new addition to the Network, application programmers with great ideas will be able to make valuable contributions to the state-of-the-art and enhance the services and pleasures of Amateur Radio packet network operation.

ROSE X.25 Packet Switch Status Update

Thomas A. Moulton, W2VY
J. Gordon Beattie, Jr., N2DSY

The Radio Amateur Telecommunications Society

INTRODUCTION

The ROSE X.25 Packet Switch has been under development for over six years. In this time, we have developed an implementation which encompasses the original design objectives, as well as requirements raised by the evolving needs of the Amateur Radio packet community. This paper describes several unique and interesting features of the Switch and introduces new features implemented in the latest release of the software.

The past two years we have seen much growth in the popularity and support for the ROSE X.25 Switch. Much of the southeastern United States, from Florida to Texas, Oklahoma to Tennessee, and on to Georgia is running ROSE. In this network, connections spanning 400 miles are not uncommon.

UPLOADABLE APPLICATIONS

The system manager may, from any point in the network, access the password protected LOADER Application. The LOADER Application allows the system manager to upload and execute a network management application program or user feature into the switch. This single concept has given the Switch the ability to support a wide range of optional features tailored to the needs of the local network environment. These include a friendly, plain-language user interface (INFO), heard list (HEARD), network connection status (USERS), Switch configuration (CONFIG), remote Switch restart (BOOTER), connection status diagnostic (UDUMP) and a CW identification which is included in the INFO Application (INFOCWID).

These applications may be loaded and deleted dynamically. For example, the CONFIG Application only needs to be loaded and present in the Switch while the configuration is being updated. After the changes to the Switch configuration have been completed, the CONFIG Application may be deleted leaving additional Switch memory free for other applications or packet buffers.

The INFO Application is provided with several optional components. These include plain-language text messages which supplement the terse status codes resident in the Switch EPROM. The messages are available in English (INFO), Spanish (INFOSP) and German (INFODE). Other languages will be offered as people volunteer to provide translations.

The INFO Application also provides a place for user accessible text which is provided by the system manager. This text may assist users with network resource information including the network addresses and callsigns of BBSs, callsign and file servers, DX Clusters and network gateways.

The newest component is an optional CWID which may be included where local regulation or custom may require its use.

HARDWARE INTERFACING

The ROSE X.25 Packet Switch may be interfaced to other ROSE Switches via Diode Matrix boards, RS-232 LAN Cards or through wire-line modems supporting standard RS-232 signals. This allows several TNCs to be grouped at a site to provide multiple ports on several RF channels.

The ROSE X.25 Packet Switch has been designed as an International Packet Network supporting a mixture of radio and landline backbone links. This is especially true when you consider the Worm-Holes that people have come up with over time. Some are simple point-to-point lease-line channels borrowed from benefactors, or X.25-based Public Data Networks or dial-up telephone connections. It is also interesting to note that with a ROSE X.25 Network, users need not know the internal network structure, so they won't really know when/if they are using a Worm-Hole!

NETWORK FAILURES

When a connection between two stations is lost (disconnected or cleared), the network sends a Clear packet to each station. This packet contains the reason for the clear and the network address that originated the clearing procedure. Most of the time the cause will be "0000" , indicating that the other station requested the clear. In this case the address will be that of the other end point.

There will be times of network congestion, or network outage where the reported address will be that of an intermediate switch. If an address of a Switch is frequently reported as detecting the failure, then it can be investigated and remedial changes may be made to the Switch or its neighbor.

A Clear packet may also occur when a connection is being set up and no operational path to the requested network end-point exists. In this case, the user will receive the network address of the primary connection setup failure.

COVERAGE

Each Switch can provide coverage to any and all local exchanges within the RF coverage of the site. What this means is that a single Switch can act as if it has many different network addresses.

CALL REDIRECTION

The Switch can be configured to trap a Call Request to a CALL @ ADDRESS and change it a a new CALL @ ADDRESS. The is useful in the case where a BBS or other network facility goes down for an extended period.

TRANSPARENT PID SUPPORT

The ROSE X.25 Packet Switch can now transparently handle ANY protocol that operates across an AX.25 connection. This means that a ROSE X.25 network connection is FULLY TRANSPARENT to the data and AX.25 PID. Specifically:

Two NOS users can establish a connection through a ROSE X.25 Network.

Two G8BPQ node users can establish a connection between them through a ROSE X.25 Network.

Two TheNET nodes can nail up a connection through a ROSE X.25 Network.

The ROSE X.25 Network doesn't care what the protocol is, it just carries the data!

When a ROSE X.25 Packet Switch receives a frame (AX.25 packet) on a user link, the PID is checked, and if it is not a regular AX.25 user (PID = F0), the PID is saved and a flag is set. The PID and data are then relayed through the network in a standard X.25 Qualified Data Packet. This Qualified Data Packet has a one byte identifier signifying that the PID is stored with the data, followed by the PID and the user data. When the frame reaches the destination Switch, it will recognize that the Data Packet contains the PID and it will use that PID when the frame is transmitted to the destination user.

CONCLUSION

The ROSE X.25 Packet Switch has undergone a significant number of enhancements that set it apart from other networking schemes. It offers flexibility for both users and system managers while simplifying the connection setup process for all. It can carry any protocol from keyboard users, BBS forwarding , TCP/IP and any new protocols that may be developed using AX.25 as the underlying Link Level protocol. Current work on an MS-DOS driver is progressing nicely. This will be a valuable extension to the network tool kit. It will allow for simplified implementation of network user and management application programmes.

THE CLOVER-II COMMUNICATION PROTOCOL

TECHNICAL OVERVIEW

Raymond C. Petit, W7GHM

ABSTRACT: This paper describes the CLOVER-II adaptive modulation controller, Reed-Solomon error-correcting coder, and selective-repeat ARQ algorithm. These operations are coordinated to obtain high-performance narrowband data communication over HF radio paths.

INTRODUCTION

The CLOVER-II strategy for sending data over the non-ideal HF radio path is to:

1. Adaptively adjust the modulation format. Slow rate modes are used to pass data in poor conditions, and high rate modes take advantage of good conditions. The channel capacity estimator provides automatic and dynamic adjustment of the modulation format for best performance in changing conditions.

2. Correct errors when possible. The large block Reed-Solomon coding system corrects errors and repairs data blocks which have been damaged by noise and signal dropouts. A finite number of errors are corrected without requiring repeat transmissions.

3. Repeat only the lost data blocks. ARQ mode only repeats those data blocks whose errors exceed the capacity of the Reed-Solomon encoder. Correctly recovered data blocks are not repeated.

The CLOVER-II carrier waveform is a composite of four tone pulses which are interleaved in time and isolated from each other in frequency. The pulse envelopes are Dolph-Chebychev functions which provide an exceptionally compact frequency spectra. The -50 dB bandwidth of a CLOVER-II signal is 500 Hz.

Data is transmitted by changing the phase and, in some modes, the amplitude of successive tone pulses. Depending on the data speed chosen, phase changes are multiples of 180 degrees (BPSM), 90 degrees (QPSM), 45 degrees (8PSM) or 22.5 degrees (16PSM). Amplitude changes are either in steps of 8 dB (8P2A) or 4 dB (16P4A).

The Reed-Solomon coder produces code blocks of size 17, 51, 85, and 255 bytes. Only the 17- and 255- length blocks are used in the ARQ protocol. A selectable parameter, code rate, sets the tradeoff between maximum number of correctable errors in a block and the overhead cost of this capacity. Higher code rates produce blocks containing larger numbers of data bytes yielding higher throughput, and fewer coder check bytes for error correction. For each block received, the coder reports the percentage of its error-correcting capacity that was required to rebuild a block or if it was overwhelmed by errors.

DATA AND CONTROL BLOCK STRUCTURE

For link maintenance, CLOVER-II modems exchange 17-byte bursts, called "CLOVER Control Blocks" or "CCB's". These are always sent in BPSM, the slowest and most robust of the modulation formats used in the ARQ protocol. Eight of the seventeen bytes are required for coding and checksum overhead for the 60% code rate. The remaining nine bytes contain control bytes, call signs, signal reports and two-way "CCB-Chat" data. A single exchange of CCB's takes exactly 2.784 seconds and is referred to as a CCB FRAME TIME, as shown at the top of Figure 1.

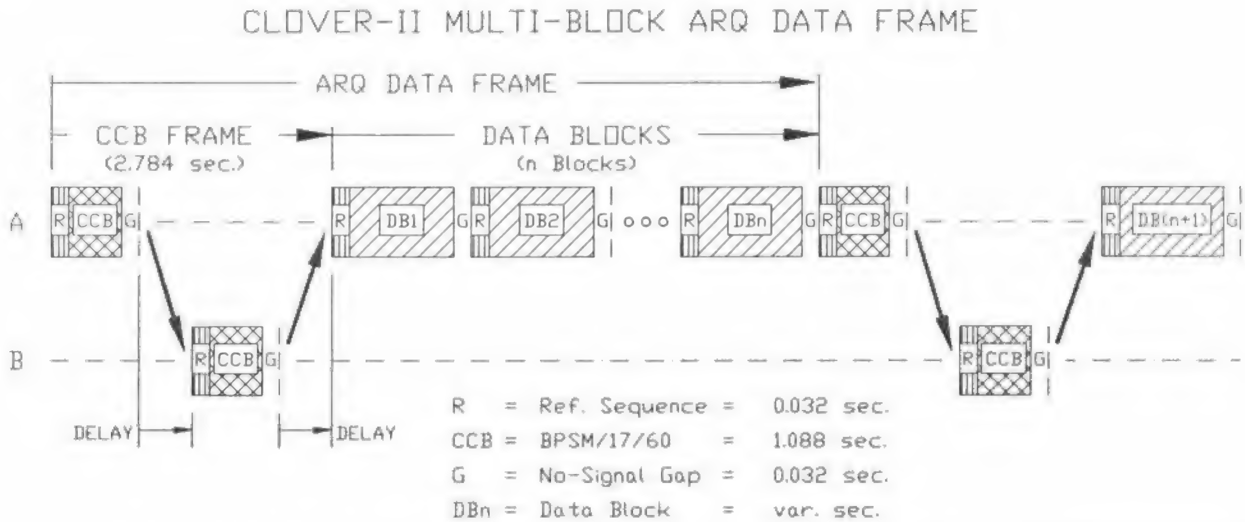
When a sending station presents a large volume of data to be transmitted, the protocol controller switches into block transfer mode. After confirmation via a CCB exchange between stations, the sending station's next CCB is preceded by a stream of one or more 255-length code blocks. The time duration of the set of 255-length data blocks is always 16.704 seconds, exactly 6 times the time required for one CCB frame (2.784 seconds). Including the CCB exchange and transmitted data blocks, each CLOVER-II ARQ frame has a length of 19.488 seconds as shown in Figure 1.

The number of code blocks sent varies with the modulation format selected by the adaptive controller. Multiple blocks are sent when fast rates are used - fewer blocks are used in slow rate mode (e.g., 6 data blocks in 16P4A format and 1 block in BPSM). The amount of data sent in each code block and the error correction capacity of the Reed-Solomon coder are set by the ARQ "Bias" parameter. The "ROBUST" bias setting (60% efficiency) provides the greatest forward error correction, but lowest total data throughput. The "FAST" bias setting (90% efficiency) provides high throughput but limited error correction. The "NORMAL" bias setting (75% efficiency) provides a compromise balance between high throughput and forward error correction. The timing and throughput of each ARQ bias setting are shown in the three tables of Figure 1.

ADAPTIVE FORMAT ADJUST

Test data has shown that measurement of the average phase dispersion of a CLOVER pulse provides a very accurate indicator of the capacity of the radio channel. While modulation produces changes in the phase of the entire CLOVER pulse, phase deviations within a pulse indicate path instability. Measurement of the detected signal-to-noise ratio (S/N) provides a secondary indicator of radio channel conditions. In typical multi-path channels, the phase dispersion can vary over several powers of 2 during intervals as short as 10 to 15 seconds. Further, it is generally the condition that S/N and therefore transmitted power is much greater than is actually required for data recovery in CLOVER.

CLOVER's estimate of current channel conditions is obtained by a process that is separate from demodulation of the data itself and is thus independent of the modulation mode in use. The phase dispersion averages are obtained over approximately 1 second samples during CCB's and 2 second samples near the end of each data block. Figure 1 shows "No-Signal Gap" (G) and "Reference Sequence" (R) portions of the CLOVER transmissions. Comparison of the receive amplitude data at these times provides a dynamic measurement of received signal-to-noise ratio (S/N = R/G) that is *not* affected by operation of radio's receiver AGC system.



ROBUST

ROBUST BIAS (60%) RATE	MOD	BLOCK	BYTES/ FRAME	MAX ERRORS	BLOCK TIME	BLKS/ FRAME	ARQ FRAME TIME	THRU-PUT BYTES/SEC
46	16P4A	255	900	300	2.720 sec	6	19.488 sec	46.2
30	16PSM	255	600	200	4.080 sec	4	19.488 sec	30.8
30	8P2A	255	600	200	4.080 sec	4	19.488 sec	30.8
23	8PSM	255	450	150	5.440 sec	3	19.488 sec	23.0
15	QPSM	255	300	100	8.160 sec	2	19.488 sec	15.4
8	BPSM	255	150	50	16.320 sec	1	19.488 sec	7.7

NORMAL

NORMAL BIAS (75%) RATE	MOD	BLOCK	BYTES/ FRAME	MAX ERRORS	BLOCK TIME	BLKS/ FRAME	ARQ FRAME TIME	THRU-PUT BYTES/SEC
58	16P4A	255	1128	186	2.720 sec	6	19.488 sec	57.9
39	16PSM	255	752	124	4.080 sec	4	19.488 sec	38.6
39	8P2A	255	752	124	4.080 sec	4	19.488 sec	38.6
29	8PSM	255	564	93	5.440 sec	3	19.488 sec	28.9
19	QPSM	255	376	62	8.160 sec	2	19.488 sec	19.3
10	BPSM	255	188	31	16.320 sec	1	19.488 sec	9.7

FAST

FAST BIAS (90%) RATE	MOD	BLOCK	BYTES/ FRAME	MAX ERRORS	BLOCK TIME	BLKS/ FRAME	ARQ FRAME TIME	THRU-PUT BYTES/SEC
70	16P4A	255	1356	72	2.720 sec	6	19.488 sec	69.6
46	16PSM	255	904	48	4.080 sec	4	19.488 sec	46.4
46	8P2A	255	904	48	4.080 sec	4	19.488 sec	46.4
35	8PSM	255	678	36	5.440 sec	3	19.488 sec	34.8
23	QPSM	255	452	24	8.160 sec	2	19.488 sec	23.2
12	BPSM	255	226	12	16.320 sec	1	19.488 sec	11.6

Figure 1. CLOVER-II ARQ Data Block Timing

Under very stable conditions on an operating radio frequency that is near the MUF, measured phase dispersion is well correlated to the signal/noise ratio, and to laboratory measurements for constant signals in additive white Gaussian noise (AWGN). In this case, adaptive mode selection is made by comparing measured phase dispersion with mode threshold values obtained from laboratory measurements under controlled S/N conditions.

When HF path parameters are "less-than-ideal" and vary widely and rapidly, the adaptive adjustment algorithm uses a more cautious approach. Although the goal is always to maximize data throughput, parameter settings which may be "optimum" at one instant may become totally unsuitable only a few seconds later. Adaptive changes may be made rapidly over several modes when path variations are well behaved. However, a less aggressive approach is required under less stable conditions. The CLOVER bias command sets three different levels of adaptive control. The bias settings are called FAST, NORMAL, and ROBUST. BIAS affects the following parameters:

1. Reed-Solomon coder "efficiency":
   FAST = 90% for high throughput but low correction capacity
   NORMAL = 75% for moderate throughput and correction
   ROBUST = 60% for high correction but low throughput

2. Phase dispersion averaging period:
   FAST = short period average for fast response
   NORMAL = moderate averaging period
   ROBUST = long averaging period to smooth wide variations

3. Modulation format selection criteria:
   FAST = favors high data rates for a given amount of phase dispersion
   NORMAL = standard mode vs phase dispersion relationship
   ROBUST = favors lower data rates for a given amount of phase dispersion

SELECTIVE REPEAT

Although CLOVER-II includes Reed-Solomon forward error correction, there are of course finite limits to the number of errors that may be corrected by the coder. In this case, CLOVER uses ARQ repeat transmission of the damaged data blocks in a similar manner to that used in AMTOR and packet radio. In the simplest form (e.g., AMTOR or packet with MAXFRAME = 1), no new data is sent until the defective block has been successfully relayed and acknowledged. This imposes no special flow management problems at either the transmitter or receiver.

The disadvantage of this scheme is that the ARQ link must be turned around after every block to obtain the acknowledgement and this imposes a high overhead cost in throughput. When conditions are good, the throughput can be increased significantly in packet radio by setting MAXFRAME to values higher then 1. The problem is that a checksum failure in an early block of the transmission will force repeat of that block *and all following blocks* of that transmission in accordance with AX.25 protocol.

Like packet, the CLOVER-II system sends multiple data blocks, but also allows *selective repeat of only the damaged data blocks.* This avoids the inefficient repeat transmission of data blocks which have already been successfully received. Adding selective repeat also increases the complexity of both transmit and receive data buffers. In CLOVER, it is further complicated by the necessity to encode all transmit data in finite sized Reed-Solomon data blocks.

The CLOVER selective repeat algorithm:

1. Buffers seven data blocks of data to be sent
2. Sending station announces data blocks to be sent
3. Receiving station acknowledges announcement
4. Sending station transmits data blocks (6 for 16P4A to 1 for BPSM)
5. Receiver buffers received data blocks, noting failures
6. Receiver passes to PC successful blocks up to 1st failure
7. Receiver requests repeat of failed block(s)
8. Transmitter clears TX buffer of successful blocks
9. Transmitter shifts blocks to be repeated to top of queue
10. Transmitter loads new data into balance of TX buffer
11. Return to step 2 and continue until complete message is sent

ON THE AIR PERFORMANCE

A few preliminary tests of the above-described protocol have been made on the air at the time of this writing. File transfers were made between AK0X and W7GHM (Boulder, Colorado to Oak Harbor, Washington, about 1500 miles) at speeds ranging from 20 to 40 bytes per second depending on conditions, in the 40, 30 and 20 meter bands. Details of subsequent testing will be given in the verbal presentation of this paper.

THE GERMAN (CENTRAL EUROPEAN) PACKET RADIO NETWORK: AN OVERVIEW

Wolf-Henning Rech, N1EOW, DF9IC @ DB0GV.DEU.EU, Pariser Gasse 2, D-6103 Griesheim, Germany
Johannes Kneip, DG3RBU @ DB0RGB.DEU.EU, Tassiloweg 3, D-8400 Regensburg, Germany

1. Fundamentals: frequency allocations and BAPT regulations

Packet radio operation in Europe made its beginnings in larger scale in the mid-80s. First access was made in the two-meter-band, but soon the single available channel (144,675 MHz) used to be overcrowded in densely populated areas. Because there is significantly less space for amateur radio frequencies in Europe our packet radio network was successfully estabilshed in the 70cm and 23cm band during the last 8 years. Tab. 1 summarizes an overview of German frequency allocation for amateur radio and its packet radio usage between 50 and 2000 MHz; regulations in other European countries is similar but more restricted at most.

One of the main reasons for the controlled growth of a homogeneous packet radio network can be seen in the German amateur radio regulations for automatic (unattended) radio station (like repeater, packet radio nodes, radio beacon or similar). The responsible amateur needs a special permission issued to radio clubs (not to individuals). The German Bundespost / BAPT as the telecom authorities will not accept any request for a node permission if the recommendation from the DARC (German Amateur radio club) or another radio club is missing; so all nodes have to be coordinated.

This procedure shows pros and cons. In the past the process often was accompanied by severe discussions with demotivated (future) node sysops waiting for their permission for many months or even years. Additional interference came from the necessity for the consent of military authorities which own the 24cm band as primary users.

But central coordination also has a large advantage: the realization of a common and homogeneous network concept instead of single nodes or regional node clusters. This conceptual work was not born by a single mind, but has been created on a couple of sysop and user meetings in 1986/87, and has since been adapted to the changing needs and perspectives of packet radio operation. As a result of this procedure it is now accepted by a majority of active amateurs who fill the concept with life.

The network described in the following sections exists (with minor modifications) in all Germany, Switzerland, The Netherlands, Belgium and Luxembourg, parts of France and Austria; Italy, Slovenia, Czeckoslovakia, Poland, Denmark and Great Britain have access to it.

2. The network structure: exclusive duplex links between nodes

When network planning started in 1986 we had the chance to use the experience of our own experiments and of other packet radio networks.

The idea of a client-server relationship between user and dedicated network suggests a separation of user access and network backbone to different communication channels.

The simple single backbone channel suffers from the same problems as the single channel user network, as there are collisions by dead-time and hidden-station effects between several nodes. The only chance to estabilsh collision free communication paths within the network is the installation of an exclusive channel for each node-to node link, either in time division by assigned time slots or in frequency division by different channel frequencies.

Band	Frequency/MHz	Status	PR channels	PR usage
6m	not available	-	-	-
2m	144-146	exclusive	3 simplex 25 kHz	no network access
1.5m	not available	-	-	-
70cm	430-440	primary	11 simplex 25 kHz 10 duplex 25 kHz (7.6 MHz shift)	network access network access
33cm	not available	-	-	-
24cm	1240-1300	secondary	19 duplex 50 kHz (59 MHz shift) 14 duplex 50 kHz (28 MHz shift)	backbone links network access

Tab. 1: Frequency allocations in Germany

The frequency division approach seems to be preferable because of at least two reasons (maybe also because it matches better the traditional way of amateur radio thinking):

- the realisation of frequency normals of a certain stability is simplier than that of synchronized time bases at different sites
- narrow continuously available channels allow transmission with less energy per bit than time shared broadband channels

Any independent traffic over several frequency channels at one site normally implies the use of different frequency bands to avoid interference between the transceivers. Such a coarse frequency separation may be realized for user access and backbone in general, but not between the individual radio links for several reasons:

- a large number of links (up to 10 from one site)
- the large-area coordination of a limited number of exclusive frequency bands is very difficult
- the same hardware should be usable for all link transceivers

A possible alternative solution is the choose of one band for all links with two well separated sub-bands in combination with the classification of all nodes in one of two classes: one group transmitting in the lower sub-band (class A node) and another using the higher sub-band (class B node). This allows collision-free links between A and B nodes, but none between nodes of the same class - a restriction that is acceptable.

From these basics a concept was born that uses the 430-440 MHz band for user access and the 1240-1300 MHz band for links, the last one with two 1 MHz wide sub-bands at the band limits (1240-1241 MHz, 1299-1300 MHz). Each node-to-node link is estabished on an individual frequency pair spaced by 59 MHz; all links use highly directional antennas which can be realizied in the 24 cm band with small dimesions and low output powers ranging from 0.5-10 W to minimize any interference.

Such links are free of collisions and therefore allow high data throughput at relative low data rates. The radios operate simplex with tx/rx shift ("half duplex") or full-duplex using independent receivers and

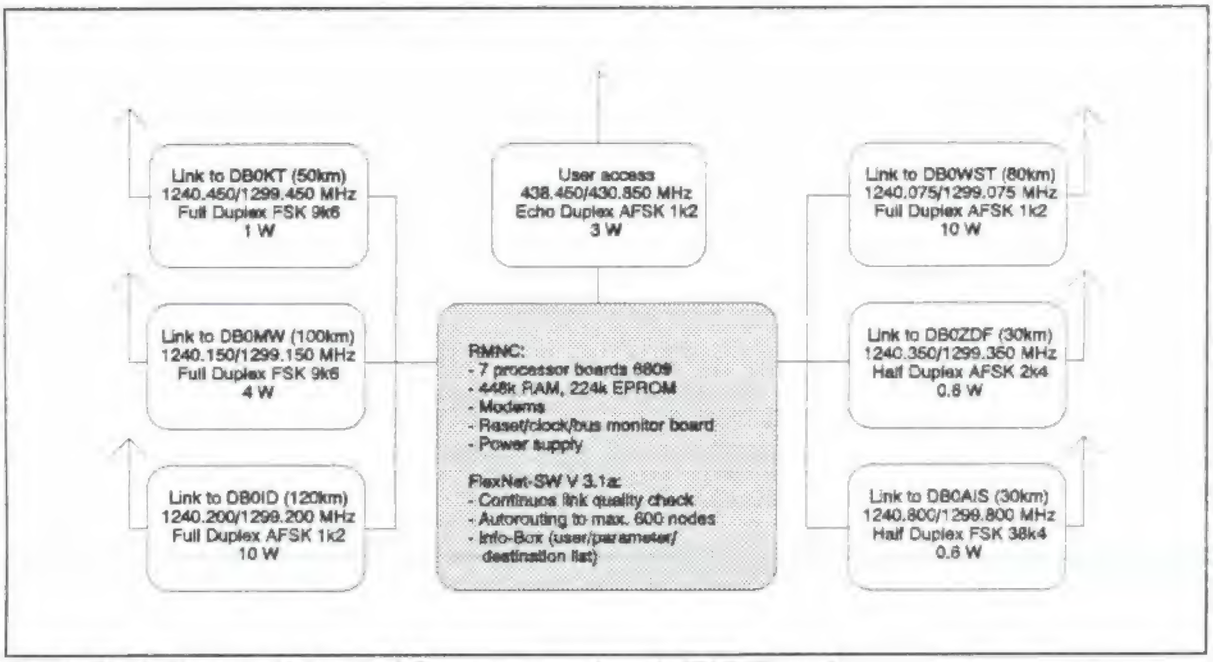


Fig. 1: Block diagram of a typical German network node (RMNC type)

transmitters and duplex filters. A network node consists of one multichannel node controller, one or two user access radios and 2...10 link radios. A typical node of this type (DB0DA near Frankfurt) is shown in a block diagram in Fig. 1.

3. The node hardware: controller and radios

The described multichannel approach needs multichannel node controllers. German PR network started using a cluster of TNC-2 connected via serial interface with the well known diode matrix.

But soon a special hardware has been designed using a parallel bus system for interchannel communication within the node. It consists of simple 6809-based processor boards, each providing one 8530 as HDLC controller. With 64kB RAM and 32kB EPROM resources are limited but sufficient because of the high parallelity (a node with 10 channels has then 640 kB for data, the code located in EPROM). This node controller system is called RMNC (Rhein-Main-NC) and is now wide-spread in Central Europe.

In addition, other systems have been developed: the (NETROM)-TNC cluster has often been replaced by a hybrid system of TNC-2s with one Atari ST or IBM PC as routing processor connected by a serial ring daisy-chaining the boards. But this hardware suffers from low data throughput on the ring at higher radio data rates.

Single IBM-PCs with SCC slot cards are also used either with TCP/IP packages or a node version of the well-known BayCom program, but their number is small.

For the moment new node controllers are under design, e.g. a V25 and a 68302 based system, both with specialized DMA capable HDLC-I/Os. Their use is still restricted because of the continuing software development.

The radios used are partly Japanese manufactured voice NBFM radios, partly surplus professional equipment, and partly home-brewed transceivers. The user access (430 MHz) is often duplex (repeater like) with digital

echo for collision avoidance; duplexers and independent rx/tx circuits are needed then. The links (1240/1299 MHz) can be half or full duplex. For that purpose several crystal controlled 1.2-GHz radio kits have been designed by one of the authors (DF9IC) which are wide-spread (many hundred units installed). Either AFSK (1k2 and 2k4), direct FSK (9k6), scrambled FSK (G3RUH type, 9k6 to 38k4), AQAM (9k6) or AQPSK (4k8) are used as modulation type.

Future work will focus on the use of higher frequencies and higher data rates. Recently the first user access points have been set up in the 1240 MHz band which will operate 9k6 or 19k2 scrambled FSK and might support in future 32k or 64k QPSK. The new links will move to the 6-cm and 3-cm microwave bands because all 19 24-cm link channels are nearly occupied. Twice 20 MHz have been reserved for packet radio links on each band using the same duplex concept, allowing data rates above 100kb/s to 1 Mb/s.

4. The node software: NETROM/TheNetNode, FlexNet, others

The available node software depends often on the type of hardware used. Hardware independence is difficult to achieve if small processor systems are used which need careful software optimization for high speed.

NETROM on Z-80 based TNC-2s is now mostly replaced by better systems. TheNetNode has more features and gives better results, using one routing processor with more ressources. But this software does not support well the node type used with its many backbone channels.

The FlexNet software developed by DK7WJ makes use of the 6809 RMNC system described above. Its actual version 3.1 supports automatic link quality test measuring continously the response time of a link and network-wide stable auto-routing based upon this link quality. Each node administrates a complete list of all other known nodes together with their estimated response time as the sum of the measured link quality on the optimum route. This information is forwarded to the neighbour nodes with a special internode communication protocol. The node supports up to 76,8 kb/s per port, and about 300 kb/s totally. The software kernel (written in C) is also ported to other platforms like 68332 but these implementations are not yet available for general use.

A similar router is also used in a node version of the BayCom software (DL8MBT) running on IBM PCs with 8530 slotcards. Its speed performance is inferior, but it has some additional features due to the available mass storage medium, as e.g. routing down to the user.

For the mentioned V25 hardware a node software with compatible router is under development also but not yet fully operating.

5. BBSs and DX clusters: a network service function

Following the given concept all stations with service functions, running unattended round the clock and handling large data volumes, like BBSs, DX clusters or TCP/IP servers, are seen as a part of the network equivalent to network nodes. They have access to it via the same collison-free links as used for internode backbone. Often they are located at the same site together wit a node with user access and are connected to it by means of a wire link.

BBSs are running with the DF3AV software on IBM compatible PCs, serving more than 25 users at the same time. S&F is taking place completely within the network, using the same communication channels as the user traffic does. Some other BBS software is also in use, often running on UNIX workstations together with the KA9Q package. Few BBS have access to additional shortwave ports to allow world wide data transfer.

DX clusters are also frequently used by shortwave and VHF DXers. Through their link feature info from whole central Europe is available in each cluster.

6. The result: a powerful and flexible amateur network

The German PR network consists of about 210 nodes, using 650 exxclusive links for backbone communication, 50 BBS and 20 DX clusters are on the air. The total number of users is between 5.000 to 10.000, but with different levels of activity.

Each BBS distributes 100...300 Mb/month to its users. S&F of private mails takes place network-wide within one hour, direct response times of some seconds per digitpeating node are achieved. During rush hour some nodes run more than 200 user QSOs at the same time (most of them on the backbone). A big node with 10 links, six of them with 9600 Baud, handles up to 8 Mbyte/h.

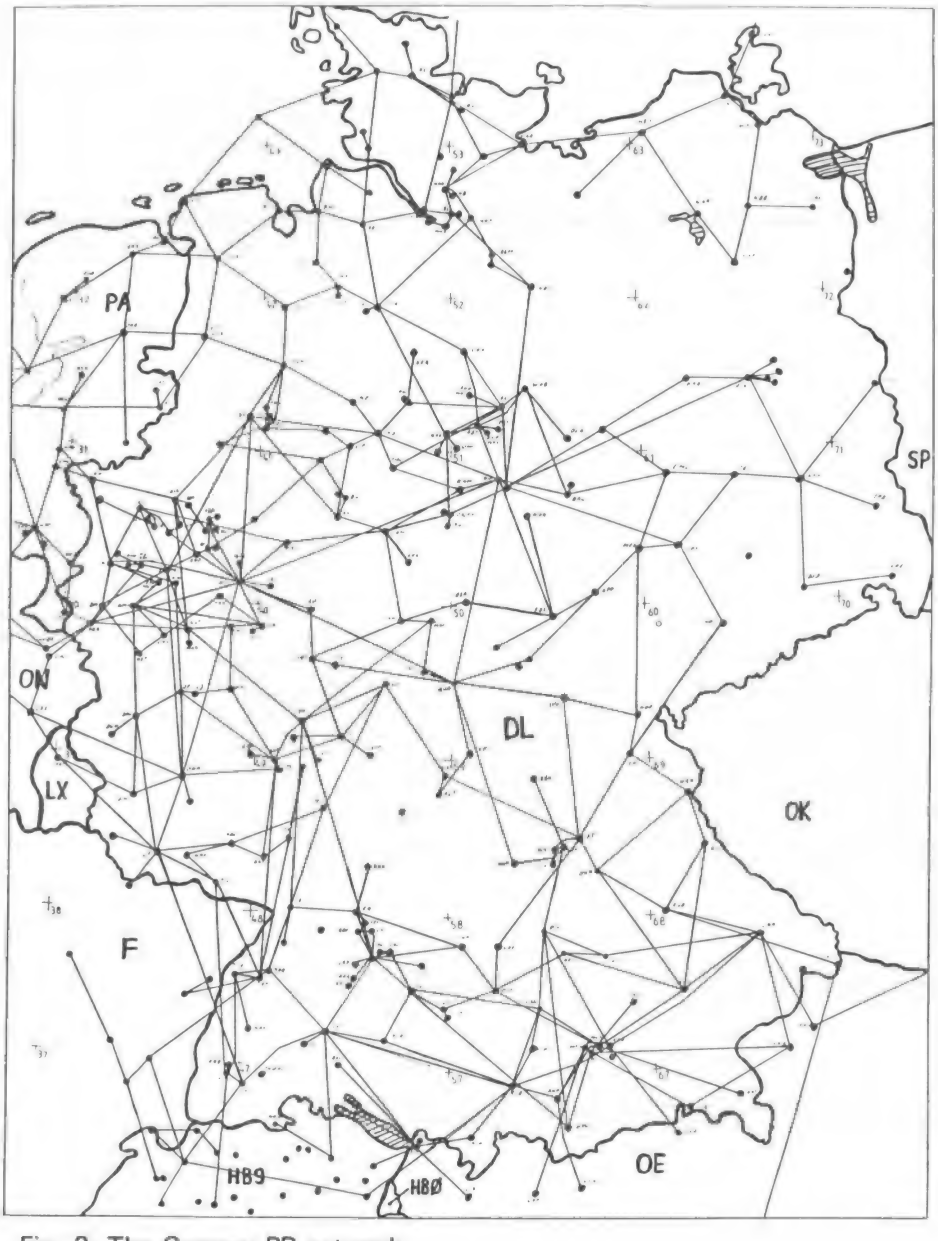


Fig. 2: The German PR network

THE GERMAN (CENTRAL EUROPEAN) PACKET RADIO NETWORK: AN OVERVIEW

Wolf-Henning Rech, N1EOW, DF9IC @ DB0GV.DEU.EU, Pariser Gasse 2, D-6103 Griesheim, Germany
Johannes Kneip, DG3RBU @ DB0RGB.DEU.EU, Tassiloweg 3, D-8400 Regensburg, Germany

1. Fundamentals: frequency allocations and BAPT regulations

Packet radio operation in Europe made its beginnings in larger scale in the mid-80s. First access was made in the two-meter-band, but soon the single available channel (144,675 MHz) used to be overcrowded in densely populated areas. Because there is significantly less space for amateur radio frequencies in Europe our packet radio network was successfully estabilshed in the 70cm and 23cm band during the last 8 years. Tab. 1 summarizes an overview of German frequency allocation for amateur radio and its packet radio usage between 50 and 2000 MHz; regulations in other European countries is similar but more restricted at most.

One of the main reasons for the controlled growth of a homogeneous packet radio network can be seen in the German amateur radio regulations for automatic (unattended) radio station (like repeater, packet radio nodes, radio beacon or similar). The responsible amateur needs a special permission issued to radio clubs (not to individuals). The German Bundespost / BAPT as the telecom authorities will not accept any request for a node permission if the recommendation from the DARC (German Amateur radio club) or another radio club is missing; so all nodes have to be coordinated.

This procedure shows pros and cons. In the past the process often was accompanied by severe discussions with demotivated (future) node sysops waiting for their permission for many months or even years. Additional interference came from the necessity for the consent of military authorities which own the 24cm band as primary users.

But central coordination also has a large advantage: the realization of a common and homogeneous network concept instead of single nodes or regional node clusters. This conceptual work was not born by a single mind, but has been created on a couple of sysop and user meetings in 1986/87, and has since been adapted to the changing needs and perspectives of packet radio operation. As a result of this procedure it is now accepted by a majority of active amateurs who fill the concept with life.

The network described in the following sections exists (with minor modifications) in all Germany, Switzerland, The Netherlands, Belgium and Luxembourg, parts of France and Austria; Italy, Slovenia, Czeckoslovakia, Poland, Denmark and Great Britain have access to it.

2. The network structure: exclusive duplex links between nodes

When network planning started in 1986 we had the chance to use the experience of our own experiments and of other packet radio networks.

The idea of a client-server relationship between user and dedicated network suggests a separation of user access and network backbone to different communication channels.

The simple single backbone channel suffers from the same problems as the single channel user network, as there are collisions by dead-time and hidden-station effects between several nodes. The only chance to estabilsh collision free communication paths within the network is the installation of an exclusive channel for each node-to node link, either in time division by assigned time slots or in frequency division by different channel frequencies.

Band	Frequency/MHz	Status	PR channels	PR usage
6m	not available	-	-	-
2m	144-146	exclusive	3 simplex 25 kHz	no network access
1.5m	not available	-	-	-
70cm	430-440	primary	11 simplex 25 kHz 10 duplex 25 kHz (7.6 MHz shift)	network access network access
33cm	not available	-	-	-
24cm	1240-1300	secondary	19 duplex 50 kHz (59 MHz shift) 14 duplex 50 kHz (28 MHz shift)	backbone links network access

Tab. 1: Frequency allocations in Germany

The frequency division approach seems to be preferable because of at least two reasons (maybe also because it matches better the traditional way of amateur radio thinking):

- the realisation of frequency normals of a certain stability is simplier than that of synchronized time bases at different sites
- narrow continuously available channels allow transmission with less energy per bit than time shared broadband channels

Any independent traffic over several frequency channels at one site normally implies the use of different frequency bands to avoid interference between the transceivers. Such a coarse frequency separation may be realized for user access and backbone in general, but not between the individual radio links for several reasons:

- a large number of links (up to 10 from one site)
- the large-area coordination of a limited number of exclusive frequency bands is very difficult
- the same hardware should be usable for all link transceivers

A possible alternative solution is the choose of one band for all links with two well separated sub-bands in combination with the classification of all nodes in one of two classes: one group transmitting in the lower sub-band (class A node) and another using the higher sub-band (class B node). This allows collision-free links between A and B nodes, but none between nodes of the same class - a restriction that is acceptable.

From these basics a concept was born that uses the 430-440 MHz band for user access and the 1240-1300 MHz band for links, the last one with two 1 MHz wide sub-bands at the band limits (1240-1241 MHz, 1299-1300 MHz). Each node-to-node link is estabished on an individual frequency pair spaced by 59 MHz; all links use highly directional antennas which can be realizied in the 24 cm band with small dimesions and low output powers ranging from 0.5-10 W to minimize any interference.

Such links are free of collisions and therefore allow high data throughput at relative low data rates. The radios operate simplex with tx/rx shift ("half duplex") or full-duplex using independent receivers and

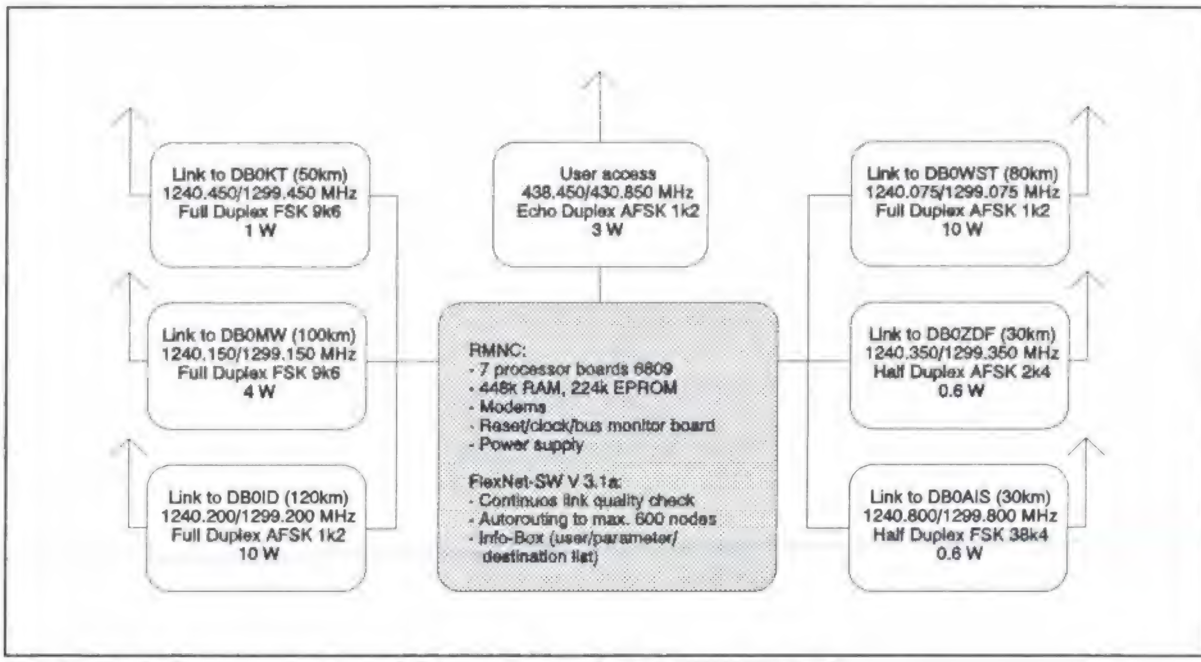


Fig. 1: Block diagram of a typical German network node (RMNC type)

transmitters and duplex filters. A network node consists of one multichannel node controller, one or two user access radios and 2...10 link radios. A typical node of this type (DB0DA near Frankfurt) is shown in a block diagram in Fig. 1.

3. The node hardware: controller and radios

The described multichannel approach needs multichannel node controllers. German PR network started using a cluster of TNC-2 connected via serial interface with the well known diode matrix.

But soon a special hardware has been designed using a parallel bus system for interchannel communication within the node. It consists of simple 6809-based processor boards, each providing one 8530 as HDLC controller. With 64kB RAM and 32kB EPROM resources are limited but sufficient because of the high parallelity (a node with 10 channels has then 640 kB for data, the code located in EPROM). This node controller system is called RMNC (Rhein-Main-NC) and is now wide-spread in Central Europe.

In addition, other systems have been developed: the (NETROM)-TNC cluster has often been replaced by a hybrid system of TNC-2s with one Atari ST or IBM PC as routing processor connected by a serial ring daisy-chaining the boards. But this hardware suffers from low data throughput on the ring at higher radio data rates.

Single IBM-PCs with SCC slot cards are also used either with TCP/IP packages or a node version of the well-known BayCom program, but their number is small.

For the moment new node controllers are under design, e.g. a V25 and a 68302 based system, both with specialized DMA capable HDLC-I/Os. Their use is still restricted because of the continuing software development.

The radios used are partly Japanese manufactured voice NBFM radios, partly surplus professional equipment, and partly home-brewed transceivers. The user access (430 MHz) is often duplex (repeater like) with digital

echo for collision avoidance; duplexers and independent rx/tx circuits are needed then. The links (1240/1299 MHz) can be half or full duplex. For that purpose several crystal controlled 1.2-GHz radio kits have been designed by one of the authors (DF9IC) which are wide-spread (many hundred units installed). Either AFSK (1k2 and 2k4), direct FSK (9k6), scrambled FSK (G3RUH type, 9k6 to 38k4), AQAM (9k6) or AQPSK (4k8) are used as modulation type.

Future work will focus on the use of higher frequencies and higher data rates. Recently the first user access points have been set up in the 1240 MHz band which will operate 9k6 or 19k2 scrambled FSK and might support in future 32k or 64k QPSK. The new links will move to the 6-cm and 3-cm microwave bands because all 19 24-cm link channels are nearly occupied. Twice 20 MHz have been reserved for packet radio links on each band using the same duplex concept, allowing data rates above 100kb/s to 1 Mb/s.

4. The node software: NETROM/TheNetNode, FlexNet, others

The available node software depends often on the type of hardware used. Hardware independence is difficult to achieve if small processor systems are used which need careful software optimization for high speed.

NETROM on Z-80 based TNC-2s is now mostly replaced by better systems. TheNetNode has more features and gives better results, using one routing processor with more ressources. But this software does not support well the node type used with its many backbone channels.

The FlexNet software developed by DK7WJ makes use of the 6809 RMNC system described above. Its actual version 3.1 supports automatic link quality test measuring continously the response time of a link and network-wide stable auto-routing based upon this link quality. Each node administrates a complete list of all other known nodes together with their estimated response time as the sum of the measured link quality on the optimum route. This information is forwarded to the neighbour nodes with a special internode communication protocol. The node supports up to 76,8 kb/s per port, and about 300 kb/s totally. The software kernel (written in C) is also ported to other platforms like 68332 but these implementations are not yet available for general use.

A similar router is also used in a node version of the BayCom software (DL8MBT) running on IBM PCs with 8530 slotcards. Its speed performance is inferior, but it has some additional features due to the available mass storage medium, as e.g. routing down to the user.

For the mentioned V25 hardware a node software with compatible router is under development also but not yet fully operating.

5. BBSs and DX clusters: a network service function

Following the given concept all stations with service functions, running unattended round the clock and handling large data volumes, like BBSs, DX clusters or TCP/IP servers, are seen as a part of the network equivalent to network nodes. They have access to it via the same collison-free links as used for internode backbone. Often they are located at the same site together wit a node with user access and are connected to it by means of a wire link.

BBSs are running with the DF3AV software on IBM compatible PCs, serving more than 25 users at the same time. S&F is taking place completely within the network, using the same communication channels as the user traffic does. Some other BBS software is also in use, often running on UNIX workstations together with the KA9Q package. Few BBS have access to additional shortwave ports to allow world wide data transfer.

DX clusters are also frequently used by shortwave and VHF DXers. Through their link feature info from whole central Europe is available in each cluster.

6. The result: a powerful and flexible amateur network

The German PR network consists of about 210 nodes, using 650 exxclusive links for backbone communication, 50 BBS and 20 DX clusters are on the air. The total number of users is between 5.000 to 10.000, but with different levels of activity.

Each BBS distributes 100...300 Mb/month to its users. S&F of private mails takes place network-wide within one hour, direct response times of some seconds per digitpeating node are achieved. During rush hour some nodes run more than 200 user QSOs at the same time (most of them on the backbone). A big node with 10 links, six of them with 9600 Baud, handles up to 8 Mbyte/h.

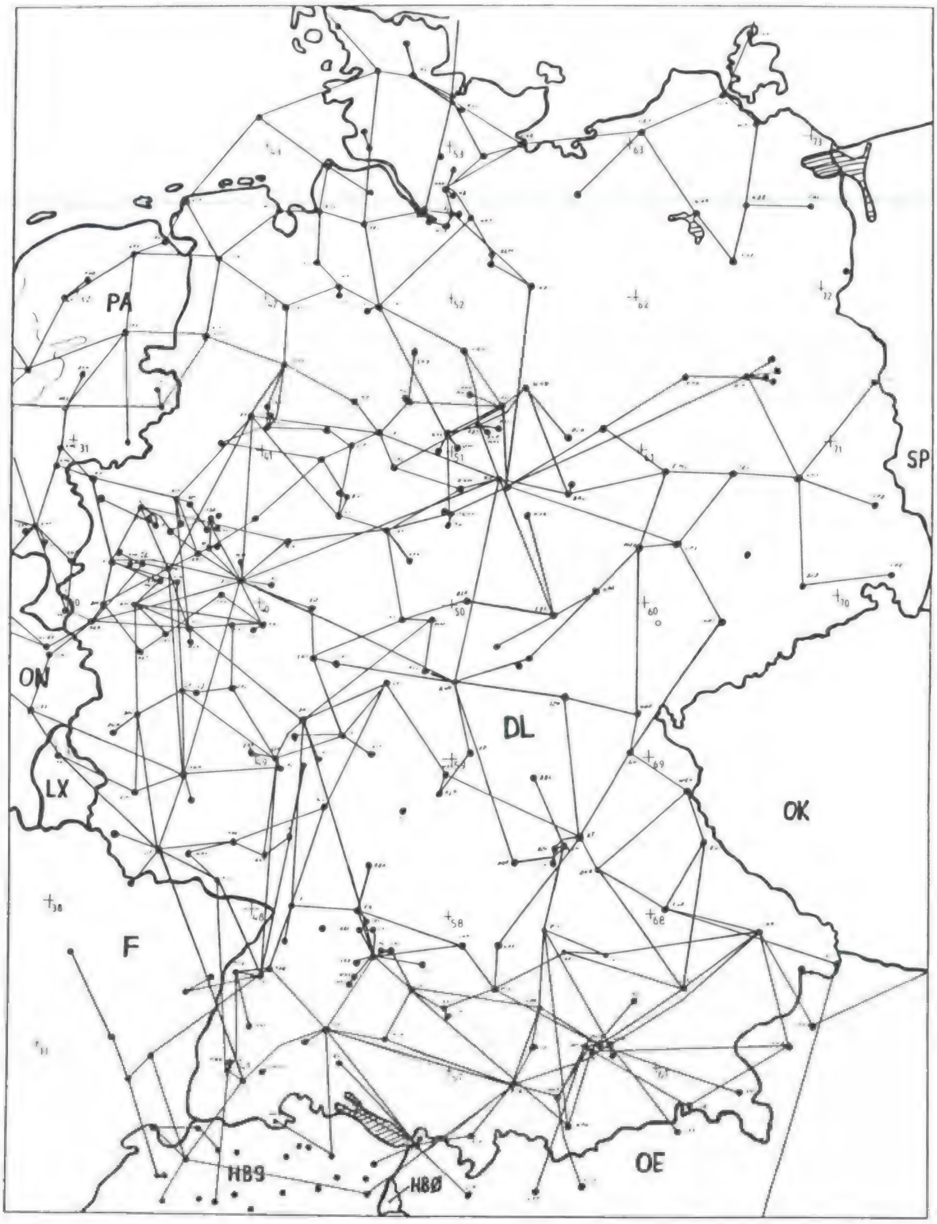


Fig. 2: The German PR network

PACTOR:
An Overview of a New and Effective HF Data Communication Protocol

Gwyn Reedy, W1BEL

Data communication via amateur radio using HF frequencies has recently become more effective and enjoyable due to a new communication protocol called PACTOR. PACTOR was developed by two enterprising German amateurs, DL6MAA and DF4KV. This article is based on the information provided by these gentlemen in their various writings.

PACTOR Features

PACTOR was designed to overcome the shortcomings exhibited by both packet and AMTOR in HF operation while remaining affordable for the average amateur operator.

- Error-free data transmission (less than 1 x 10-5)
- True binary data transmission
- Efficient use of channel capacity
- Good interference tolerance
- Requires only 600 Hz channel bandwidth
- Complete visibility of sending and receiving callsigns

Why PACTOR?

HF propagation is characterized by multipath propagation which induces 'bit stretching' and phase distortions, fading, impulse noise, and interference by other stations, among other obstacles to communication.

The PACTOR mode is similar to AMTOR which is good for ordinary HF communication. Both use half duplex ARQ; packets (blocks) of data carrying the information are acknowledged with short 'receipt' signals by the receiving station. When errors occur, the receiver can request the repetition of a packet with relevant control signals.

PACTOR uses a MASTER/SLAVE phasing like AMTOR. The SLAVE clock is synchronized to the MASTER timing. Only the MASTER corrects his Receive phase.

A long series of tests conducted by DL6MAA and DF4KV have shown that for operation in rapidly changing conditions, it is not a good policy to adjust packet length automatically. Simulations and on-the-air testing showed the optimum HF packet length to be about one second. To compensate for varying conditions, PACTOR varies the number of data characters in the data block, but does not change any of the synchronous timing parameters. PACTOR determines the proper baud rate to use based on the accuracy of bit transitions and the link error statistics.

Data blocks have CRC-16 checking as is done in AX.25 packet. This is much more robust than the parity bits FEC used in AMTOR.

The data field of the PACTOR packet can contain any digital information; the format of the codes is specified in the status byte. At the present time the choice is between 8 bit ASCII and Huffman compressed 7 bit ASCII.

Authorization

The US FCC regulations specify that Baudot or ASCII codes may be used for data transmission. The 8 bit ASCII text transmitted by PACTOR is closer to 'pure ASCII' than the bit-stuffed HDLC used by AX.25 packet, so there should be no question of the legality of this mode. The compressed PACTOR data mode uses ASCII characters encoded in a channel-capacity-enhancing format which still meets the intent of the regulations. The regulations regarding amateur transmissions require that there be no intent to obscure the data transmitted. The Huffman encoding scheme is published as an appendix to this article. It seems reasonable to me to consider that encoding

for spectrum efficiency using a widely published encoding scheme shows no intent to encrypt the content of the data transmission and is therefore allowable under US regulations.

Performance

PACTOR achieves good throughput during poor HF conditions by a variety of techniques. The actual baud rate is kept low - the same as AMTOR. This is one third the rate of typical HF AX.25 packet operation. From another point of view, AMTOR and PACTOR bits are three times as long as 300 baud HF packet bits, thus providing much increased protection from the bit smearing caused by multipath propagation.

A doubling of the throughput compared to AMTOR results from sending longer blocks of data (but still short enough to cope with most fades) thus reducing the percentage of overhead carried. In addition, the ability to automatically double the data content of each block under favorable conditions provides a considerable increase in efficiency.

Finally, encoding ASCII text (7 bit characters) using Huffman codes increases throughput by an average of at least 70 percent.

Memory-ARQ

A significant feature of PACTOR is Memory-ARQ. Copies of the repeated reception of the same packet which fails the CRC are aggregated in memory and are summed individually for each bit. The aggregate of all unsuccessful transmissions is decoded which effectively increases the signal to noise ratio by about 15 dB. This PACTOR feature is hardware dependent and prevents the proper implementation of PACTOR as a software- only upgrade to packet or AMTOR equipment.

The combination of the above factors provides a protocol which can provide a throughput nearly equal to HF packet in the best of conditions, and much better throughput than packet during typical conditions. Compared to AMTOR, the throughput in good conditions is up to four times as great. During the poorest of conditions, throughput is considerably better than AMTOR because of the CRC-16 error checking and Memory-ARQ capabilities.

Appendix: PACTOR Huffman Code

Huffman coding is relatively indifferent to differences between real and theoretical alphabet character frequencies, so that similar good results are obtained in German and English plain text. The compression factor attained with ASCII amounts to about 1.7, resulting in an average of 4.5 bits per character. A greater compression factor would require considering the statistical relationships between the individual characters (Markov encoding).

Code in order of frequency, LSB (sent first) on the left:

Character	ASCII	Huffman
space	32	10
e	101	011
n	110	0101
i	105	1101
r	114	1110
t	116	00000
s	115	00100
d	100	00111
a	97	01000
u	117	11111
l	108	000010
h	104	000100
g	103	000111
m	109	001011
<CR>	13	001100
<LF>	10	001101
o	111	010010
c	99	010011
b	98	0000110
f	102	0000111
w	119	0001100
D	68	0001101
k	107	0010101
z	122	1100010
.	46	1100100
,	44	1100101

S	83	1111011	'	39	110001101110
A	65	00101001	_	95	111100001100
E	69	11000000	&	38	111100111001
p	112	11000010	+	43	111100111110
v	118	11000011	>	62	111100111111
0	48	11000111	@		0001010111000
F	70	11001100		36	0001010111001
B	66	11001111	<	60	0001010111010
C	67	11110001	X	88	0001010111011
I	73	11110010	#	35	0010100011011
T	84	11110100	Y	89	00101000110101
O	79	000101000	;	59	11110000110100
P	80	000101100		92	11110000110101
1	49	001010000	[	91	001010001101000
R	82	110000010	]	93	001010001101001
(	40	110011011		127	110001101111000
)	41	110011100	~	126	110001101111001
L	76	110011101	}	125	110001101111010
N	78	111100000		124	110001101111011
Z	90	111100110	{	123	110001101111100
M	77	111100110	'	96	110001101111101
9	57	0001010010	^	94	110001101111110
W	87	0001010100	<US>	31	110001101111111
5	53	0001010101	<GS>	29	111100001101100
Y	121	0001010110	<ESC>	27	111100001101101
2	50	0001011010	<EM>	25	111100001101110
3	51	0001011011	<CAN>	24	111100001101111
4	52	0001011100	<ETB>	23	111100001110000
6	54	0001011101	<SYN>	22	111100001110001
7	55	0001011110	<NAK>	21	111100001110010
8	56	0001011111	<DC4>	20	111100001110011
H	72	0010100010	<DC3>	19	111100001110100
J	74	1100000110	<DC2>	18	111100001110101
U	85	1100000111	<DC1>	17	111100001110110
V	86	1100011000	<DLE>	16	111100001110111
<FS>	28	1100011001	<RS>	30	111100001111000
x	120	1100011010	<SI>	15	111100001111001
K	75	1100110100	<SO>	14	111100001111010
?	63	1100110101	<FF>	12	111100001111011
=	61	1111000010	<VT>	11	111100001111100
q	113	1111010110	<HT>	9	111100001111101
Q	81	1111010111	<BS>	8	111100001111110
j	106	00010100110	<BEL>	7	111100001111111
G	71	00010100111	<ACK>	6	111100111000000
-	45	00010101111	<ENQ>	5	111100111000001
:	58	00101000111	<EOT>	4	111100111000010
!	33	11110011101	<ETX>	3	111100111000011
/	47	11110011110	<STX>	2	111100111000100
*	42	001010001100	<SOH>	1	111100111000101
"	34	110001101100	<NUL>	0	111100111000110
	37	110001101101	<SUB>	26	111100111000111

LOW COST ENTRY INTO PACKET RADIO USING DIGICOM
BY CHRISTOPHER C. RENDENNA, KB2BBW

The purpose of this paper is to provide information on setting up a low cost packet radio system via the Digicom modem and software. It assumes the reader is minimally computer literate and has a basic knowledge and understanding of packet radio basics.

Florian Radlherr (DL8MBT) can be called the father of DIGICOM, although DL2MDL and DL3RDB must be credited as well. Digicom is a software-based packet radio system for either the C64 or the C128 developed in the mid 1980's. The Digicom system comprises of 1) modem 2) software 3) computer and 4) radio.

The modem utilizes the AM7910 chip which allows both VHF and HF operation. Some of the older modems use the TCM-3105 chip The modem is available from various distributors and clubs (See Appendix A) assembled or in kit form at reasonable prices. I have supplied schematics (See Appendix B) for those wishing to build their own modem. The majority of the parts are available from local electronic stores while the chips can be mail ordered from larger distributors. Check your yellow pages.

In a recent phone call to Commodore Business Machines, they indicated that approximately 10 million C64's and another 7 million C128's have been manufactured and sold. These units are not only affordable compared to the IBM's and clones, but are a common hamfest item at bargain prices. The C64 is recommended for its inexpensiveness, however, its 40 column screen is a primary drawback. The user may not be aware of his 40 column handicap in a realtime packet QSO. However, once a BBS is accessed or file transfers are engaged where columns and listings are specifically aligned, the user may have visual difficulty equating 1 line of 80 columns with 2 lines of 40. Some Digicom software versions incorporate a hi-res option that will allow the user to switch from 40 column to 80 column mode. Although the intentions of this feature are good, it is impractical. One would need a magnifying glass to actually comprehend anything. To make the text easier to read, the user can obtain a green or amber screen Composite Monitor in the 12" range.

An important note regarding the new "flatbed style" C64C computers have a ceramic capacitor soldered across the cassette motor line. The motor line is used for TX in both Digicom and Digiprom versions. The capacitor is marked on the C64C as C84 and must be clipped out. Removal of this capacitor will not affect the normal usage of the computer.

A 1541 disk drive is used to load the floppy. Users

have daisy chained these drives to increase capacity while others have successfully hooked up Commodore hard drives. Those unable to acquire or afford a disk drive can purchase the software in cartridge form. Although not available for all programs, the cartridge versions are identical to the disk versions.

The C128 is the alternate option to the C64. The primary disadvantage is price. With a TV set as a monitor, the computer will display 40 columns in C128 or C64 mode. The back of the computer will allow an additional RGB monitor by way of a 9 pin cable. This will enable the user 80 columns in C128 mode, and thus, run C128 Digicom software without the inconvience of overlapped lines as discussed above.

Computer die-hards with Commodore +4's and SX-64's will be pleased to find out there still is use for their equiptemnt as well. The Commodore +4 requires a memory add-on to operate the software written for it. The SX-64 has no cassette port, so the modem must be installed inside the computer. (See Appendix A for modification source)

The software is the workhorse for the system. It is the result of hard work on the part of the Digicom team in Germany and the Digiprom team in the United Kingdom. All software is public domain (shareware) and may be copied and distributed as long as all files are kept entact and no fees are charged. Unscrupulous entrepeneurs who have attempted to monopolize on these copyrighted versions have realized their folly after being taken to court! Over the years many versions of software have been circulated, some being US modifications. (See Appendix C) Operation of the latest version will ensure the user of a bug free version and latest features. It is recommended that if the user wishes to run a NODE, use Digicom V3.60 or Digiprom V1.00, to run a PBBS (mailbox) use Digiprom V4.07A (UK version has no third party mail), to run a PBBS (mailbox) with auto-forwarding/reverse forwarding use Digiprom MB-XA. As of this paper, the latest Digicom Version 5.0 has been released from Germany.

The user's first task is to identify the version being used, load up your favorite word processor and print out the documents. Those anxious to get started immediately, may begin the version simply by typing LOAD"*",8,1. I have found that several versions, after being loaded, will not execute when typing RUN at the ready prompt. Since the software is written is machine language, typing the command SYS30720 after the ready prompt will start the program. The time should be entered initially, then the user's callsign with the command MYCALL <callsign>. All commands are preceded by a colon (:). Lines without a colon indicate converse mode and will be transmitted upon a carraige

return. Selection between 300 baud HF and 1200 baud VHF operation is done through a switch on the modem, not the software. Any user attempting to receive packet signals at HBAUD 1200 setting should experiment with values between 1135 and 1175. Also, make sure audio output on the modem is high enough by listening to yourself on a comparable receiver if connects are not being made.

There are two amazing basic features to the software that have always intrigued me. First, programs of the same format can be transferred between two computers without loading any special programs or transfer protocols. Second, the software allows remote control operation with varying degree of security levels. This is a feature that enables other users to do something as simply as changing the hosts screen color, to more complex activities such as writing data files or programs to disk. With a hardware add-on, relays can be remotely turned on and off by packet. In addition to these basic features, some specialized versions include personal connect messages, mailbox options, autoforward and autoreverse (done with permission from your local packet BBS SYSOP), Local Area Network configuration (excellent to use when set-up between multiple users - it also allows for by-pass of major nodes eleviating congestion), and many more.

Finally, most radios are easily wired to the modem with little or no modification. Appendix D contains the wiring diagrams for the most popular HT's and mobiles.

Feel free to write to any of the amateurs in the source list for help. But please read the manuals first. Digicom is an excellent low cost system that will give you hours of enjoyment!

APPENDIX A

Drude, Ted KA9ELV. "My SX-64 Runs Digicom". 73 Amateur Radio, October 1989.

JSM Group. "Digiprom V4 Series Instructions". Peterborough, England.

Kutner, Barry N., M.D. W2UP. "Digicom>64 - A Software-based Packet Radio System for the Commodore 64". 73 Amateur Radio, August 1988.

ACKNOWLEDGEMENTS

Schematics courtesy of A&A Engineering, Anaheim, CA.

Thanks to Bill G6WWW of the non-profit Digiprom team who supplied which much need information.

SOFTWARE SOURCES

John Bearsby. PO Box 500, GPO Mirrabooka 6061. West Australia.

Don Henry. 21 Hillview Drive. Aldavilla 2440. NSW. Australia.

Archie Janks. PO Box 37126. Birnam Park. J/Burg 2015. South Africa.

Con Kapoutsis. 78 South Worple Way. Mortlake, London SW14 8NG. Great Britain.

Peter Mallett. 22 Meehan Street. Blenheim 7301. South Island. New Zealand.

Micheal Mullikin, 5626 Viking Drive, Jackson, MI 49201.

Chris Rendenna, KB2BBW 709 Ten Eyck Avenue, Floor 2, Lyndhurst, NJ 07071.

Rick Silverio, c/o C.A.R.S., PO Box 653, Meadville, PA 16335.

Jack Thomas. 1705 East 28th Street 17. Ashtabula, OH 44004.

Lars Thunberg. Box 109. S-640 31 Melloesa. Sweden.

HARDWARE

A&A Engineering, 2521 West LaPalma, Unit K, Anaheim, CA 92801. (714) 952-2114

Crawford Amateur Radio Society, PO Box 653, Meadville, PA 16335.

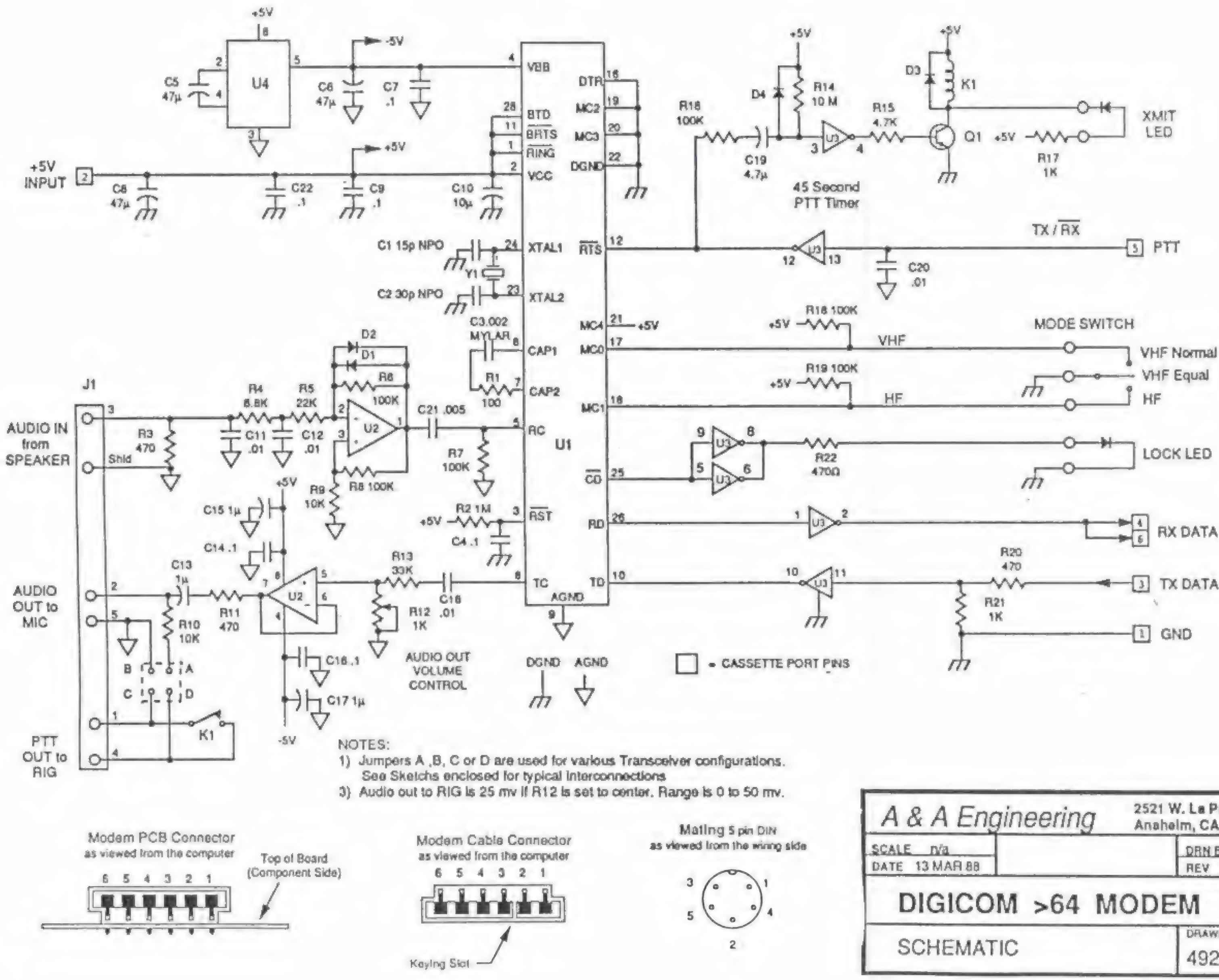
EasyTech, 2917 Bayview Drive, Freemont, CA 94538. (800) 582-4044. (One source for TCM-3105 chips.)

Ramsey Electronics, Inc., 793 Canning Parkway, Victor, NY 14564. (716) 924-4560.

USER SUPPORT

The DIGICOM EXCHANGE, Chris Rendenna, Editor. 709 Ten Eyck Avenue, Floor 2, Lyndhurst, NJ 07071 (201) 939-6986.

This monthly newsletter was started on a voluntary basis over three years ago. Send SASE for latest issue.



NOTES:
1) Jumpers A ,B, C or D are used for various Transceiver configurations. See Sketchs enclosed for typical interconnections
3) Audio out to RIG is 25 mv if R12 is set to center. Range is 0 to 50 mv.

A & A Engineering		2521 W. La Palma, Unit K Anaheim, CA. 92801 USA
SCALE n/a		DRN BY STAS A
DATE 13 MAR 88		REV 13 JAN 92
DIGICOM >64 MODEM		
SCHEMATIC	DRAWING NUMBER 492-154	REV C

APPENDIX C
USERS SOFTWARE GUIDE

The following is an updated list originally published in THE DIGICOM EXCHANGE Vol. 2 No. 2 Feb. 1991. This revised list contains information obtained from Mike (KF6PU), Con (G6URT), Bill (G6WWW), Gorch (DF3MH) and Juan (LU3AGC).

DC>64 V1.42 - Utilizes User port. Least powerful of all versions. Has no multiconnect or binary xfer. Protype.

DC>64 V1.50 - This is the original version written by DL8MBT in 1985 and was donated to the amateur world FREE.

DC>64 V1.51 - No multiconnect or binary xfer. Has gateway capability if two C-64s and two modems are used. (Not an official release)

DC>64 V1.52 - Same as V1.51 but with callsign W2UP in header. Hbaud corrections must be made with each of these versions as 50 hz clock is used. HB 1200 (1156) and HB 300 (289). (Not an official release.)

DC>64 V2.00 and DC>128 V2.00 - These are the last official versions from BAPR. Multiconnect, binary xfer, remote security levels, chat and node (true store and forward virtual circuit networking) are available.

DC>64 V2.03 - Same as V2.00 with Special character set, Status line. SABM retry bug is fixed and Dwait can be set to zero. Has Password security system and callsign and morning or evening annunciators can be stored in ctext with Ctrl-A and Ctrl-B.

DC>64 V2.10 - same as V2.00 but include improved printer and PERM routines.

DC>64 V2.1B - First of the PBBS versions of Digicom. Superceded by V3.00

DC>64 V2.11 - Great Britain revamp of V 2.00.

DC>64 V2.22 - Same as V2.10 with reverse print status line. "Port" replaces "DC-" on status line. "STBY" replaces "QRV". SABM retry bug is fixed. Dwait can be set to zero. Drives 8 and 9 can be accessed either locally or by remote. HDLC routines improved.

DC>64 V3.00 - Kantronics style PBBS functions. Mail can be read by msg. number. Messages are displayed in RLI/MBL-style format. This version dedicates Port 4 exclusively to Digicom node functions. K # and KM not supported. Sysop can not selectively delete mail. PBBS-TIDY utility provides rather crude file maintainence capability.

This is a revamp of 2.11 by a Belgian amateur.

DC>64 V3.10 - Combines a debugged V3.00 with modified V4.0 PBBS TIDY. SABM retry bug is fixed. Dwait can be set to zero when desirable. V3.10 maintains prompt structure of V3.00 which is superior to V4.00.

DC>64 V3.11 - Same as V3.00 but upgraded by the JSM Group to include CW identification to comply with British regulations.

DC>64 V4.00 - Similiar to V3.00 . Double slashbars are no longer required for remote commands. Has Hi-Res loader with sound. Pbbs and Node command prompts are combined. V4.0 PBBS TIDY is much improved over earlier version. Selective file deletion is possible as is renumbering of messages. This is the first Digiprom written by Pete (G7AMW). This was a test program that was never intended to be released.

DP>64 V4.01 - Same as V4.00 with bugs removed by author Pete (G7AMW). Rewritten by Pete (G7AMW) to remove bugs and avoid illegal profits that were being made on V4.00 by unscrupulous people. Copyright notice and name change to Digiprom made. Available on cartridge and disk.

DC>64 V3.50 - Released in November of 1989(?). Many new functions added. Autorouting, expanded Editor functions, more node connect ports, quit text, and other functions.

DC>64 V3.51 - Supposedly there is a bug in V3.50 that is fixed in V3.51. Bugs in IPOLL. With some distributed copies, however, the program will lock-up when the user attempts to send a message while in MSYS BBS'. (See Vol 3 No. 2 Feb 92 and Vol. 3 No. 4 Apr 92 DIGICOM EXCHANGE) It is suspected that V3.51A is a USA custom version, however, as DIGICOM users pass along copies, the "A" has been left off or added randomly. This makes identification very difficult.

DC>64 V3.51A - This was a modified version to V.3.51 (which in turn was a modification to V.3.50!). There were a couple of glitches, but the modifier or the origin of this is unknown. What has happened over the years is that the is "A" is sometimes dropped or added when hams make disk copies for others. The end result is that you can't be sure which of the two you are actually getting.

DC>64 V3.60 - Official DL8MBT program. Excellent features enable configuration as 1) one operating port and PBBS (Personal Bulletin Board Service ie: mailbox) or 2) four operating ports, thus utilizing the full features.

DC>64 V3.61 - Same as V3.60 but for the C16.

DP>64 V4.05D - A mailbox program similar to V3.00, but now KM is supported. Other features include Jheard, Net-Rom node

simulation, different port connect messages, and page tone. Four extensive manuals extremely similar to V4.00. Could be used as a terminal program. Some copies need to be loaded with DP, then SYS30720 to run. Update to DP V4.01.

DP>64 V4.06A - This modified version from GI0EEC attempted to clear up bugs that 4.05D had. However, this version still had bugs.

DP>64 V4.07 - Four ports supported with autoforwarding from your local BBS to the mailbox. However, bug in autoforward.

DP>64 V4.07A - Major overhaul to this version in which memory locations had to be moved. Autoforwarding bug fixed. Two versions written: one for UK without 3rd party mail capability, and one non-UK version that does support 3rd party capability.

DC>64 V5.0 - Latest from the German Digicom Group. Features

include: Simultaneously use of two Channels and higher Baud Rates up to 9600 Bd (Even 19200 Baud are possible, but only good for a C128 as the C64 has a too slow Screen output) with an SCC-Cardridge plugged into the expansion-Slot of the C64 or C128. Crossband operation with e.g. different Speeds is possible. C64 now manages 20 Ports (4 Screens and 16 Nodeports), C128 manages 40 Ports (4 Screens and 36 Nodeports). Path up to theoretic 37 Calls lenght possible. Node transparent what means lenght and PID of Frames are repeated in original form. Node answers with a little Prompt and handles now the "Recon-nected to..." feature used by other Nodes. Path-finder now only recognizes the three messages "connected", "busy" and "fail", whereby messages like "link setup..." no longer disturb the link setup procedure. Use of the Fast Serial with an 1581 Floppy Disk Drive with the C64 possible by fixing two wires. This is recommended when using higher baudrates and file-in/out. People running a too long TXDEL can become masked out by a new command (MXFLAGS). Many, many more features...

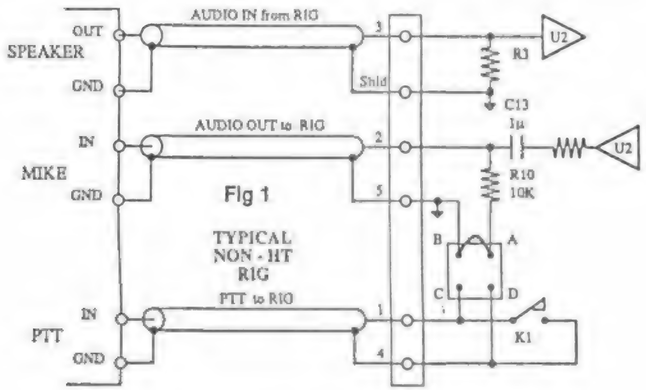
DP>64 MB-XA - Mailbox program (PBBS) with autoforwarding. Word processor included. Written for use with 801, 802, 803 and 1101 Commodore printers although some non-C64 printers will work. Meant to compliment V4.07A, 80 column mode only!

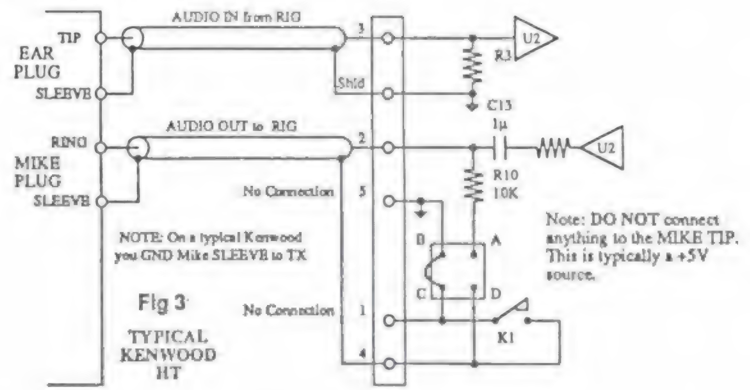
DP>128 MB-XA - Self copying (self-extracting) program written exclusively for the C128 with three independent routines that check the system. Same program as above but for C128.

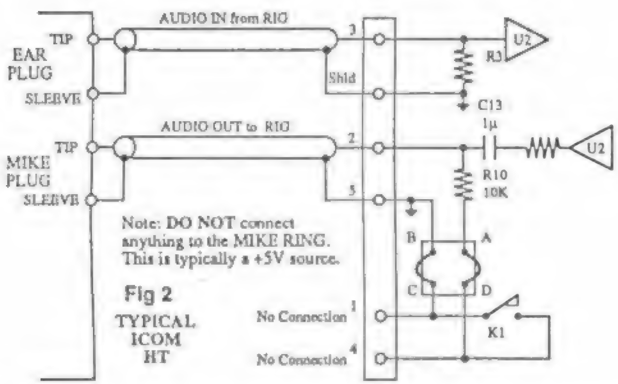
DP>64 MB-XB - Currently under development and testing.

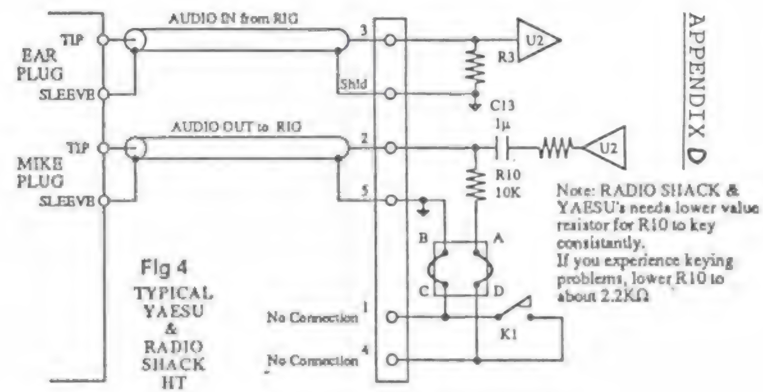
DP> NODE V1.00 - Meant to be used in conjunction with JSM Gateway and DP V4.07A, but can stand alone too. Coomands include Bye, Connect, Info, JHeard, Mail, Nodes, Users.

TYPICAL MODEM INTERCONNECTIONS







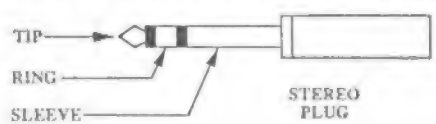


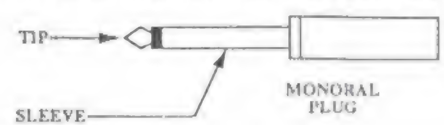
Notes:

1) Figure 1 shows a typical, Non - Handi -Talky interconnection. Three separate shielded cables are recommended. If your rig does not supply three separate grounds, you may combine the sheilds at the rig. Minature michrophone cable (lapel mike cable) is recommend.

2) Figures 2, 3 and 4 show interconnections for typical HT's. Most Handi-Talkies don't have a separate PTT line. They are put into the transmit mode by applying a DC load from the MIKE HOT to GROUND, then super-imposing (AC coupling) the audio onto this same line. R10 applies the DC load and C13 does the AC coupling. These HT interconnection diagrams were submitted to us by various manufacturer's, these interconnections do work and are in wide use. (SOME KENWOOD's ARE DIFFERENT)

3) **NOT ALL of a certain manufacturer's rigs are the same.**
YOUR RIG MAY BE DIFFERENT. CHECK YOU OPERATORS MANUAL IF YOU HAVE PROBLEMS.





A&A Engineering
2521 W. LaPalma, Unit K, Anaheim, CA 92801
Phone (714) 952-2114 FAX (714) 952-3280

Using ROSE X.25 Packet Networks

Bill Slack, NX2P
Don Rotolo, N2IRZ
Andrew Funk, KB7UV
Thomas A. Moulton, W2VY

Radio Amateur Telecommunications Society

Some find ROSE X.25 Packet Network operation a mystery. this is likely due to simply a lack of information and/or experience with this approach to packet networking.

This paper is a slightly modified version of the Users Guide distributed by the Radio Amateur Telecommunications Society (RATS) to users of the RATS-operated ROSE X.25 Packet Network. It is presented here to help familiarize others with ROSE X.25 Packet Network features and operations.

0.1 History

Tom Moulton, W2VY, wrote generic user instructions for ROSE X.25 Packet Networks which are distributed along with Switch code. As part of his efforts constructing and operating the ROSE X.25 Packet Network covering Northwestern New Jersey, Eastern Pennsylvania and Southern New York, Bill Slack, NX2P, created an excellent User Guide based upon Tom's work.

Don Rotolo, N2IRZ, expanded and modified the guide to cover the entire RATS ROSE Network. Andrew Funk, KB7UV, took this work and modified it for presentation to this conference.

[Any errors or omissions are mine. —kb7uv]

1. The ROSE X.25 Packet Network

The ROSE X.25 Network provides short and long distance connectivity, all initiated by a simple connect command at your TNC. To connect to another station, you only need to know:

The other station's callsign
The callsign of your local switch
The address of the other station's local switch[1]

This information is typed into your TNC as a normal connect command. ROSE X.25 Packet Networks "look like" a pair of intelligent digipeaters, with a callsign specifying the point you enter the network and an address specifying the point you exit the network. All of the routing from switch to switch is handled by the network, just like the telephone system.

All connects using the ROSE network are done from your TNC's `cmd:` prompt, by issuing a connect command of the following form:

C callsign Via [*entry digi,*]*switch callsign,*[*DNIC,*]*exit address*[*,exit digi*]

where:

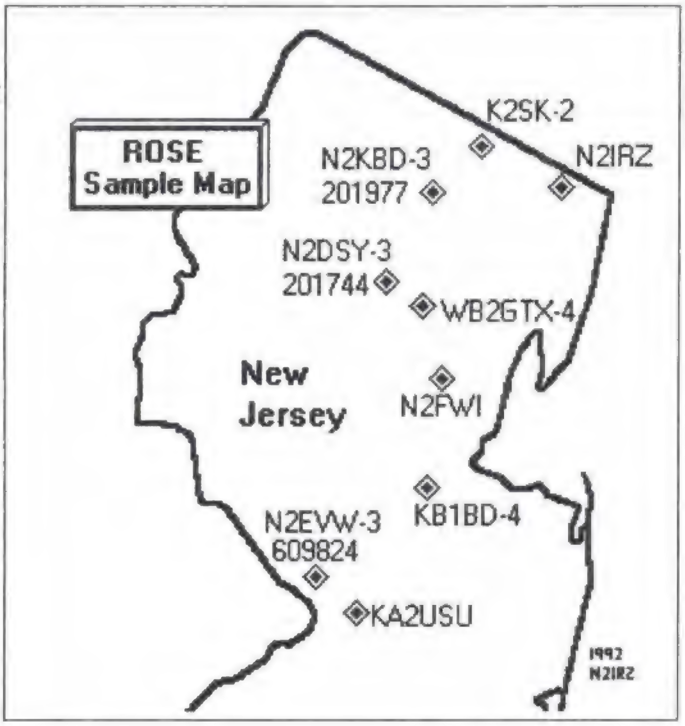
callsign	is the callsign of the station you want to connect to. This is usually an Amateur callsign, but may take other forms (such as HEARD or CROWD), and may include an SSID.
entry digi	(Optional) is the callsign of a digipeater required to access your local ROSE Switch.
switch callsign	is the callsign of your local ROSE Switch. ROSE switches do not beacon, but you may see it in use. Generally, ports for USER access to the RATS ROSE Network are on the 2m band, with a-3 SSID. Other networks may use different conventions.
DNIC	(Optional) is the four-digit Data Network Identification Code for the ROSE Switch local to the other station. This is only used when connecting into another country. A list of ROSE Data Network ID Codes is provided later in this Users Guide.
exit address	is the six-digit[2] address of the ROSE Switch local to the station you want to connect to. (In the RATS ROSE Network, addresses for a particular area code may be found by connecting to the INFO application at that area code and exchange 555. For example: 201555 for area code 201.)

exit digi (Optional) is the callsign of a digipeater required between the station you want to connect to and their local ROSE Switch. Also see entry digi.

1.1 Some Examples

As an example, we will look at how a basic connect command is made and then try a few variations. To help with these examples, we've created a make-believe network map[1]. Normally, such a map is unnecessary with ROSE networks, but in this case it will help to visualize switch locations.

My callsign is N2IRZ. Suppose I wanted to connect to my local BBS, WB2GTX-4. From the map, I see that the N2DSY-3 (201744) switch is nearest to WB2GTX-4, and on the same frequency. My local switch is N2KBD-3—I know this because I see it on the air often. Alternately, I could have found my local switch using the User Port listing that is available from RATS. So, to connect to the BBS, at my TNC's `cmd:` prompt I would issue this connect command:




```
C WB2GTX-4 V N2KBD-3,201744
```

Once N2KBD-3 acknowledged my connection, my TNC would say:

```
*** Connected to WB2GTX-4.
```

Imediately after that, the *network* would acknowledge my connect request by sending the message `Call being Setup`. I would then wait a few moments while the network set up the connection. When the connection is established, the network would tell me by sending the message:

`Call Complete to WB2GTX-4 @ 3100201744`[2]

At this point, I am connected to the BBS, and everything operates as if I were connected directly. If the connection attempt had failed for any reason, the network would inform me and provide the reason for the failure by sending a disconnect code[3]. Refer to Section 3 for more details.

Now a few variations. Suppose I was visiting a friend in Trenton, where the local ROSE Switch's callsign is N2EVW-3. To connect to WB2GTX-4, I would type:

```
C WB2GTX-4 V N2EVW-3,201744
```

at my TNC's `cmd:` prompt. Note that the only change is my entry point into the network, in this case N2EVW-3 instead of N2KBD-3. My exit point from the network (201744) as well as the callsign of the BBS both remain the same.

Now suppose that when I came back from Philadelphia, I wanted to connect to my friend for a keyboard-to-keyboard "conversation." Knowing that my friend's callsign is KA2USU, that N2EVW-3 is his local ROSE Switch, and that N2EVW-3's ROSE address is 609824, I would type:

```
C KA2USU V N2KBD-3,609824
```

Of course, my local ROSE Switch in this case is N2KBD-3.

Now suppose I wanted to connect to another friend, who lives near the N2DSY-3 (201744) ROSE Switch. I would type:

```
C N2FWI V N2KBD-3,201744
```

Compare this with the first example.

As a final example, If I again wanted to connect to WB2GTX-4, and I couldn't reach N2KBD-3 directly, I could use the K2SK-2 digipeater as an entry digipeater. In this case, I would type:

```
C WB2GTX-4 V K2SK-2,N2KBD-3,201744
```

Once again, the basic form of the connect command remains the same.

Refer to Section 1 above for the detailed syntax of a ROSE X.25 Network connect command, and remember that all connect commands to the ROSE network are made while DISCONNECTED from the local switch.

1.2 The ROSE Address

Every ROSE Switch has a unique callsign and address. The callsign is the same as any other Amateur Radio callsign as used on packet, and usually has an SSID. The address consists of ten digits (in North America), which is broken into two parts. The first four digits are the X.121 Data Network Identification Code (DNIC), which is an internationally recognized standard[6]. The last six digits are uniquely assigned to each ROSE Switch based upon location. In North America, the 3-digit telephone area code and the 3-digit telephone exchange are combined for six digits. Other countries may use different addressing

schemes, perhaps with different length addresses, as required by national standards or regulations.

If the user does not specify the DNIC when making the connect request, the network assumes that the exit address is within the country of origin. The DNIC portion of the address is not shown on the maps, since it is the same for all switches in the USA. For example, the full address of the N2DSY-3 ROSE Switch is 3100201744, where 3100 is the DNIC for the USA. If you are attempting an international connection[7] you must specify the DNIC. Note that the DNIC uses its own digipeater field, because a TNC will not allow more than 6 digits in any one field.

Now you know how the addressing works in a ROSE Switch. You may ask why an address is used at all, when the callsign is also a unique identifier. The answer is ROUTING. If callsigns were used, then each switch in the network would have to know about every other switch in the network. This addressing scheme allows a ROSE Switch to route the connect request based upon standardized information, thereby allowing for routing to a practically unlimited number of switches, locally, regionally, nationally and worldwide[8].

1.3 Entry and Exit digipeaters

The ROSE Switch allows for the optional use of one digipeater at each end of a ROSE Network connection. Both, one or neither digi may be used, as necessary. For example, say I could only reach the N2KBD-3 ROSE Switch via a digipeater, K2SK-2, and KA2USU needed the K2GL-2 digi to reach N2EVW-3. The connect command to my TNC would look like:

```
C KA2USU Via K2SK-2,N2KBD-3,609824,K2GL-2
```

As another example, suppose I wanted to connect to TIØPAQ (Chuck) in Costa Rica, again using a digipeater at each end:

```
C TIØPAQ v K2SK-2,N2KBD-3,7120,100110,TI2CES-2
```

That represents a real example of the longest possible connect command you may have to make using a ROSE X.25 Network. 7120 is Costa Rica's DNIC, 100110 is the ROSE address local to TIØPAQ, and TI2CES-2 is the digi he needs to use.

1.4 Call Progress Messages and Disconnect Codes

When you issue a connect command using the ROSE Network, messages indicating the progress of your call are sent so you know something is happening. For example, if you were to issue the following command:

```
C WB2GTX-4 Via N2DSY-3, 201744
```

N2DSY-3 would send you an acknowledgement of your connect request on behalf of WB2GTX-4. At this point your TNC's connected status LED lights, and your TNC generates the familiar `*** Connected to...` message, but this doesn't indicate that your connection to WB2GTX-4 is complete. Along with the connect acknowledgement, N2DSY-3 also sends you a message `Call being Setup`, indicating that your call has been accepted by the network and is being routed. Once the call has been completed to WB2GTX-4, N2DSY-3 sends you another message:

```
Call Complete to WB2GTX-4 @ 3100201744.
```

You are now connected to WB2GTX-4.

If for some reason the connection to the destination station cannot be made, or a disconnection occurs, your local ROSE Switch will "clear the call" and send you a code explaining the reason before actually disconnecting. One reason for a call clearing is if the other station is busy. Another reason would be a normal disconnect, such as sending "b" ("bye") to a PBBS.

These code takes the form:

`*** Call Clearing *** #### `*XXXXYYYYYY*

where #### is a four digit Hexadecimal number[9] explaining the reason, and *XXXXYYYYYY* is the DNIC and ROSE address of the Switch originating the message. Some common codes are listed here—a complete list appears later in

0000 Remote Station disconnected	Normal disconnect from other station, such as sending "b" to a PBBS
0100 Remote Station is Busy	The other station is either busy or has CONOK set OFF
0900 Link is Out if Order	One of the switches used by your connection has failed in some way and there is no alternate route available. Or, you may have entered an invalid address—check for a typo! If you think a Switch has failed please tell the Network Sysop—Often network users are the first to detect problems.
0D00 Route not Known	Either you have entered an invalid address or the Switch is not configured properly. After verifying the address, if the failure repeats alert the Network Sysop.
3900 Remote Station Not Responding	Either the station you are trying to reach is not on the air, is not hearing the Switch you specified in the exit address

Common ROSE X.25 Disconnect Codes

this Users Guide. A switch can be configured to also provide a plain-text explanation of each code, in various languages. Refer to Section 3.3.

2. Call Traceability and Accountability

One unique advantage of ROSE X.25 Packet Networks is the traceability of connections. For example, I have connected to KA2USU using this connect command: `C KA2USU Via N2KBD-3, 609824`. If I were to type the text "Hello Ted", someone monitoring 223.4 would see N2EVW-3 transmit the following frame:

```
N2IRZ>KA2USU,201977,N2EVW-3*: Hello Ted
```

First, note that the ROSE Switch always identifies its transmissions with its own callsign—never the callsign of any user. While this is a legal requirement in some countries, it also makes ID beacons (and the resultant waste of channel time) unnecessary. Second, note that each frame carries all of the information required to connect back to me. Just like any digipeater connection, you would simply reverse the order of the digipeater fields. Thus, to connect back to me after I disconnect, you could use the command:

```
C N2IRZ Via N2EVW-3, 201977
```

With the ROSE network there is never any question as to who is connected to whom, which station is transmitting, or how to reach the remote station—all that information is included with every transmitted frame.

3. ROSE Applications

The ROSE Switch supports three[10] user-accessible applications: INFO, USERS and HEARD. These applications can be optionally uploaded by the ROSE Switch sysop to provide functions which are not built into the standard ROSE Switch software. To use an application, simply connect to it. For example, to get the heard list from the Trenton, NJ ROSE Switch, you might type (assuming your local switch is N2KBD-3): `C HEARD Via N2KBD-3, 609824`. After receiving the "Call Complete" message you will receive the application's output[11]. Please see the HEARD, USERS and INFO Application instructions following for more details.
Note that, since these applications are uploadable at the sysop's option, they may not be in all Switches. If the application you are trying to connect to is not loaded into the ROSE Switch at the address you specify, you will receive a call clearing code of 3900. If you would like a particular application loaded into a switch, send a message to the ROSE network sysop.

3.1 The HEARD Application

The HEARD application is very useful when looking for stations to connect with at a remote network address. "Last Heard" lets you know how recently a station was heard, and "RXCnt" gives some insight into how reliable a path is going to be (higher RXCnts mean better paths), as well as the other station's activity level. This information makes it much easier to select a station to connect to than a simple list. To connect to HEARD, issue a command like:

```
C HEARD v Localswitch, Address
```

where Localswitch is the call of your local switch, and Address is the address of the switch you want a HEARD list from.

A sample HEARD session is shown below:

```
cmd: c heard v kb7uv-3 201744
*** CONNECTED to HEARD VIA KB7UV-3,201744
Call being Setup
Call Complete to HEARD-0 @ 3100201744
ROSE X.25 Packet Switch Version 3.1 (920911) by Thomas A. Moulton, W2VY
Heard List for N2DSY-3   3100201744
                                 Last   First  (How long ago)
Port Station    Destination Heard  Heard  RXCnt FType Path
  0  KB7UV-3    N2DSY-3     00:00  25:56   3498   I
  0  N2IRZ-3    N2DSY-3     00:00  10:27    522  RR
  1  N2DSY-6    N2DSY-3     00:00  25:59   2304  RR
  1  N2DSY-12   N2DSY-3     00:00  25:53   1952  RR
  0  KB7UV-1    HEARD       00:01  00:09     18  RR   KB7UV-3,201744
  0  HEARD      KB7UV-1     00:01  00:01      2   I   201744,KB7UV-3*
  0  N2KZH-12   WA2ERD-12   00:01  18:29    684  RR
  0  WB2GTX-4   RATS        00:02  25:56   1109  UI   N2DSY-2
  0  KB2BBW     CQ          00:03  21:00     28  UI
  0  KA2VLP-3   N2DSY-3     00:04  23:19   4940  RR
  0  N2KZH-4    PBBS        00:04  21:22   1101  UI
  0  KA2YKC-4   BEACON      00:04  25:43   1896  UI
  0  WA2ERD     BBS         00:04  04:43      7  UI

Type H to redisplay or * for ALL or Disconnect now
END>
```

Port: 0 means the Radio port, 1 means the RS–232 port (direct link to co–located switches on other frequencies)

Station: The station that sent the packet

Destination: The station that the packet is sent to

Last Heard: Hours and Minutes ago that the most recent packet from station was heard

First Heard: Hours and Minutes ago that first packet from station was heard

RXCnt: Total number of frames received from station

FType: (Frame Type) Last frame type monitored from station

Path: Lists digipeater fields used between station and destination

3.2 The USERS Application

The USERS application is useful for determining who is connected to a remote station or server (i.e., what Virtual Circuits (VCs) are passing through a switch). There are several other functions which are mainly of interest to the network sysop: the total amount of memory available and the amount in use; the connect status of each switch in a cluster; the status of each VC passing through the switch (e.g., Pending, Connected, etc.); and links status. A more detailed explanation of these parameters may be found in the ROSE System Manager's Manual[12]. To connect to USERS, issue a command like:

```
C USERS v Localswitch, Address
```

Where Localswitch is the call of your local switch, and Address is the address of the switch you want a USERS list from.

A Sample USERS list is shown below:

```
cmd: C users v n2kbd-3,201977
*** CONNECTED to USERS VIA N2KBD-3,201977
Call being Setup
Call Complete to USERS-0 @ 3100201977
ROSE X.25 Packet Switch Version 3.1 (920911) by Thomas A. Moulton, W2VY

User List for N2KBD-3   3100201977
Memory Size is:  27788 Bytes
Memory Used is:  18528 Bytes
EPROM Checksum: 26h

N2IRZ-9   X.25 Trunk (R1) with the following connections:
N2IRZ     @ 3100201790     (  1 P4 D1) -> USERS    @ 3100201977
NX2P-10   X.25 Trunk (R1) with no connections.
N2IRZ-12  X.25 Trunk (R1) with no connections.
N2KBD-6   X.25 Trunk (R1) with no connections.

There are no calls Pending.

The Following X.25 Trunks are listed as Out of Order:
<None> - All Links Operational

Type U to redisplay or Disconnect now
END>
```

The USERS list above shows only one user —N2IRZ— who is connected from the Switch at address 201790 to the USERS application at this Switch (Address 201977). The VC passes on to the N2IRZ-9 Switch. To find out where it goes

from there, connect to USERS at that Switch. The three other Switches in this cluster (NX2P-10, N2IRZ-12 and N2KBD-6) have no VCs from this Switch (201977). It is possible, however, that they are carrying VCs from other Switches — to determine that, connect to USERS at the Switch. Please note that most backbone Switches do not have applications loaded, and therefore their addresses are not shown in the User Ports and Services listing. Contact your local network sysop for more information about backbone switches.

3.3 The INFO Application

The INFO application has three functions:

- Allow users to remotely obtain a brief text file describing a particular switch, which can otherwise be obtained (without INFO) only by *directly* connecting to the Switch and pressing ENTER.
- Provide Network Services ("555") and Users ("411") Directory Servers. These services, described in detail below, help users find their way around the network.
- Adds clear-text descriptions to "Call clearing" codes (See section 3), making them easier to understand. The text descriptions are presently available in English, Spanish, and German.

Using INFO, you can retrieve text from a remote switch, in order to learn a little about it. In many cases the INFO text from a distant switch will contain information about the distant area that might otherwise be unknown.

Connecting to the INFO application is just like any other ROSE connection:

```
C INFO v Localswitch, Address
```

where Localswitch is the call of your local switch, and Address is the address of the switch you want the INFO text from.

3.3.1 The 555 Server

Every Area Code served by the RATS ROSE Network has an INFO Server providing a complete list of all User-Access ROSE Switches within that Area Code. Also listed are all locally available network services. This special INFO server responds to the address *XXX*555, where *XXX* is the 3-digit Area Code.

For example, to get the list for the 201 Area Code, issue the command:

```
C INFO via Localswitch, 201555
```

Where Localswitch is the call of your local switch.

3.3.2 The 411 Server

Similar in nature to the 555 Server, each Area Code also has a 411 Server. This application contains a list of local users and where they can be found. Stations are only listed by request, so contact your local Network Sysop to be added to the list.

For some Area Codes the 555 and 411 lists are combined into a single listing. In these cases connections to INFO at ROSE Address *XXX*411 and *XXX*555 will both respond with the combined list.

If you encounter problems accessing either of these servers, or have updated information, please contact the network sysop.

This is an example of a combined 411 and 555 listing:

```
cmd:c info v kb7uv-3, 718555

*** CONNECTED to INFO VIA KB7UV-3,718555
Call being Setup
Call Complete to INFO-0 @ 3100718555
ROSE X.25 Packet Switch Version 3.1 (920911) by Thomas A. Moulton,
W2VY
ROSE Network Backbone --Astoria, Queens-- KB7UV & RATS


               *** ROSE DIRECTORY BULLETIN ***
                   Area Codes 718 and 212
                       Update 02/21/92

Note: Link to POLI/NOAA/NWS 212 switch not yet in place... Stay tuned!

Callsign  Address  Type     Name            Alias    Hours
--------- -------- -------- ---------       -------- --------
KB7UV-4   718956   BBS      Andy Funk       BBS      24 Hrs
WB2GTX-4  718204   BBS      PARC                     24 Hrs
K2ULR-15  718204   BBS      CBS SFX ARC              24 Hrs

For INFO on other AREA Codes in the Network (currently
201,908,609,914,215)
use ROSE output destination 201411, 908411, 609411, etc.

If you wish to be added to this (718) list please contact Andy, KB7UV.
```

```
     Switches Available for User Access
   in the 718 Area Code As of 01/14/92 are:

Address Callsign  Location     User Port Freq
------- --------  -----------  --------------
718204  KB7UV-3   Astoria, NY  145.07 Mhz

     Services Available for User Access
   in the 718 Area Code As of 06/17/92 are:

Address Callsign  Alias   Location       Service
------- --------  -----   -------------
------------------------------
718204  WB2GTX-4          Secaucus, NJ   ROSErver/PRMBS BBS
718956  KB7UV-4   BBS     Astoria, NY    ROSErver/PRMBS BBS,
Multi-User

For Info on Switches and Services Available in other Area Codes in
the Network, currently 609,908,201 use ROSE output destination
609555, 908555, or 201555

Address questions about the KB7UV Packet Services, via packet
radio
mail, to KB7UV@KB7UV.#NLI.NY.USA

This switch brought to you courtesy of the Radio Amateur
Telecommunications Society (RATS).  For information on RATS
address
packet mail to "ASKRAT@KB4CYC.NJ.USA".

73, Andy, KB7UV


Please Disconnect now
```

4. Further Information

Additional information on ROSE X.25 Packet Networking can be found in:

- ROSE X.25 Packet Switch System Managers' Manual
- ROSE X.25 Packet Switch Resource Manual

These documents, and the executable files for the ROSE X.25 Packet Switch, ROSErver/PRMBS Packet Radio MailBox System, ROSErver/OCS Online Callbook Server, ROSE/RZ network maintenance utility, ROSE/STS Station Traffic System for managing NTS traffic, and ROSE/RMAILer PBBS Remote Mail Server, are all available from the Radio Amateur Telecommunications Society (RATS). Please include an SASE with all inquires.

Correspondence may be sent to:

RATS
PO Box 93
Park Ridge, NJ 07656-0093

Via the Internet, RATS can be reached at address:

`rats@kb2ear.ampr.org`

Packet inquires may be sent to:

`askrat@kb4cyc.nj.usa`

Voice inquires can be directed to Nancy, N2FWI, and Gordon, N2DSY, Beattie. Their number is 201-387-8896.

Software and support is available on the RATS KB7UV Landline ROSErver/PRMBS. The system supports data rates of 1200 to 9600 bps (V.32), and J-, X-, Y-, and Zmodem binary protocols. It can be reached at 718-956-7133. Callers should wait for the "login:" prompt (don't even press ENTER!) and follow the instructions provided.

5.0 ROSE X.25 Call Clearing Codes

Every time a a call is cleared, the ROSE X.25 Packet Network provides a code indicating the reason. The code is a 4-digit hexadecimal number, where the last two digits are always 00. These codes are the universally accepted X.25 Cause Codes standardized by CCITT.

Number	CCITT X.25 Name	Explanation (ROSE X.25 Useage)
0000	DTE Originated	The other station disconnected (normal disconnect)
0100	Number Busy	The other station is busy, or has CONOK set OFF
0300	Invalid Facility	Internal network error—notify Network Sysop!
0500	Network Congestion	Retry count exceeded
0900	Out of Order	Network link not operating
0B00	Access Barred	Cannot connect to a network trunk
0D00	Not Obtainable	No known path for address specified
1100	Remote Procedure	Internal network error
1300	Local Procedure	internal network error
1500*	RPOA Out of Order	(not used)
1900*	Reverse Charge	(not used)
2100*	Incompatable Dest.	(not used)
2900*	Fast Select	(not used)
3900	Ship Absent	No response from other station
C100*	Gateway Proc. Error	(not used)
C500*	Gateway Congestion	(not used)

* Currently not used, should not be seen.

6. X.121 Data Network Identification Codes (DNIC)

Zone 2

2020	Greece
2040	Netherlands
2060	Belgium
2080	France
2120	Monaco
2140	Spain
2160	Hungary
2180	E. Germany*
2200	Yugoslavia
2220	Italy
2260	Romania
2280	Switzerland
2300	Czechoslovakia
2320	Austria
2340	Great Britain and Northern Ireland
2380	Denmark
2400	Sweden
2420	Norway
2440	Finland
2500	USSR*
2600	Poland
2620	Germany (W)*
2660	Gibraltar
2680	Portugal
2700	Luxembourg
2720	Ireland
2740	Iceland
2760	Albania
2780	Malta
2800	Cyprus
2840	Bulgaria
2860	Turkey

Zone 3

3020	Canada
3080	St. Pierre and Miquelon
3100	United States
3300	Puerto Rico
3320	US Virgin Islands
3340	Mexico
3380	Jamaica
3400	French Antilles
3420	Barbados
3440	Antigua
3460	Cayman Islands
3480	British Virgin Islands
3500	Bermuda
3520	Grenada
3540	Montserrat
3560	St. Kitts
3580	St. Lucia
3600	St. Vincent
3620	Netherlands Antilles
3640	Bahamas
3660	Dominica
3680	Cuba
3700	Dominican Republic
3720	Haiti
3740	Trinidad & Tobago
3760	Turks & Caicos Is.

Zone 4

4040	India
4100	Pakistan
4120	Afghanistan
4130	Sri Lanka
4140	Burma
4150	Lebanon
4160	Jordan
4170	Syrian Arab Rep.
4180	Iraq
4190	Kuwait
4200	Saudi Arabia
4210	Yemen (Arab Rep.)*
4220	Oman
4230	Yemen (Dem Rep.)*
4240	United Arab Emirates
4250	Israel
4260	Bahrain
4270	Qatar
4280	Mongolia
4300	UAE (Abu Dhabi)
4310	UAE (Dubai)
4320	Iran
4400	Japan
4500	Korea
4520	Vietnam
4540	Hong Kong
4550	Macao
4560	Democratic Kampuchea
4570	Laos
4600	China
4700	Bangladesh
4720	Maldives

Zone 5

5020	Malaysia
5050	Australia
5100	Indonesia
5150	Phillipines
5200	Thailand
5250	Singapore
5280	Brunei
5300	New Zealand
5350	Guam
5360	Nauru
5370	Papua New Guinea
5390	Tonga
5400	Solomon Islands
5410	New Hebrides
5420	Fiji
5430	Wallis & Futuna Is.
5440	American Samoa
5450	Gilbert and Ellice Is.
5460	New Caledonia & Dep.
5470	French Polynesia
5480	Cook Islands
5490	Western Samoa

Zone 6

6020	Egypt
6030	Algeria
6040	Morocco
6050	Tunisia
6060	Libya
6070	Gambia
6080	Senegal
6090	Mauritania
6100	Mali
6110	Guinea
6120	Ivory Coast
6130	Upper Volta
6140	Niger
6150	Togolese Republic
6160	Benin
6170	Mauritius
6180	Liberia
6190	Sierra Leone
6200	Ghana
6210	Nigeria
6220	Chad
6230	Central African Republic
6240	Cameroon
6250	Cape Verde
6260	Sao Tome and Principe
6270	Equatorial Guinea
6280	Gabon Republic
6290	Congo
6300	Zaire
6310	Angola
6320	Guinea-Bissau
6330	Seychelles
6340	Sudan
6350	Rwanda
6360	Ethiopia
6370	Somali Dem. Rep.
6380	Rep. of Djibouti
6390	Kenya
6400	Tanzania
6410	Uganda
6420	Burundi
6430	Mozambique
6450	Zambia
6460	Madagascar
6470	Reunion
6480	Zimbabwe
6490	Namibia
6500	Malawi
6510	Lesotho
6520	Botswana
6530	Swaziland
6540	Comoros

Zone 7

7020	Belize
7040	Guatemala
7060	El Salvador
7080	Honduras
7100	Nicaragua
7120	Costa Rica
7140	Panama
7160	Peru
7220	Argentina
7240	Brazil
7300	Chile
7320	Columbia
7340	Venezuela
7360	Bolivia
7380	Guyana
7400	Ecuador
7440	Paraguay
7460	Suriname
7480	Uruguay

* Given recent political changes, it is advised to confirm the proper DNIC with local authorities.

7. Notes:

1. In a properly configured ROSE X.25 Packet Network, the address of the other station's local ROSE Switch is likely to be the other station's telephone Area Code and exchange.

2. In North America, switch addresses consist of six digits—the telephone area code and 3-digit exchange. In other countries the addressing scheme may differ. Some TNCs, as well as some other networking systems, will not accept an all-numeric digipeater field. The ROSE Switch permits you to substitute the letter O for a zero and either L or I for a one in the address.

3. The sample map is used only for this example. Contact RATS for accurate network maps.

4. The 3100 part of the address shown is the X.121 Data Network Identification Code (DNIC) for the United States. Please refer to Section 1.2 for more information about the DNIC

5. This will appear in the form of a 4 digit number in Hexadecimal. A properly configured ROSE Switch will also give you a brief text explanation. A complete listing of the codes, which are internationally standardized CCITT X.25 disconnect codes, is given later in this User Guide.

6. A complete listing of standard X.121 Data Network Identification Codes is given in Section 6 of this User Guide.

7. Not possible at this time, but soon, as Central America and Australia both have extensive ROSE Networks.

8. This may be expanded to the known universe, when necessary.

9. These codes are standard CCITT X.25 Cause Codes. The last two digits are always zero.

10. As of this writing. Other applications are being developed.

11. Prior to version 2.8, HEARD and USERS waited for the user to press before sending their data.

12. The ROSE System Manager's Manual is being rewritten at this time (9/92). Release is expected 10/92.

Sophisticated Mail User Interface Systems
Keith Sproul, WU2Z
Mark Sproul, KB2ICI

Radio Amateur Telecommunications Society

Keith Sproul, WU2Z
Magnolia Road
North Brunswick, NJ 08902
AppleLink: Sproul.K
Internet: wu2z@sproul.com

Mark Sproul, KB2ICI
1368 Noah Road
North Brunswick, NJ 08902
AppleLink: Sproul.M
Internet: sproul@sproul.com

ABSTRACT

Most of us use many different electronic mail systems such as the Internet, Usenet, Compuserve™, America-Online™, Genie™, Prodigy™, and of course Amateur Radio Packet Mail. Most of these E-Mail systems have had some sort of sophisticated user interface developed for them. For Usenet, there are many different News Reader programs. Compuserve has CIS Navigator and CIS Information Manager. Many of us also use some sort of Local Area Network-based E-Mail systems at our work. LAN-based E-Mail systems such as Microsoft Mail™ are extremely sophisticated, especially those that run on windowing environments such as the Macintosh™ and Microsoft Windows™.

With very few exceptions we all get paper mail delivered to our homes. We don't have to go to the post office to pick it up each day. The current packet mail systems require that we go to the post office (BBS) each day and ask for our mail. We also must ask for a list of public messages and explicitly specify which ones we want to see. Then we wait for 1200 baud data to come across the busy air waves.

A sophisticated user interface that would take care of retrieving a user's messages and specified bulletin topics would be a significant improvement. Such a system should deliver the mail that a user wants to his desktop, without the user having to go and get it manually. Such a system may also be expanded to then forward the mail to a user via another mail system so that the individual uses only one program at his computer.

The authors have developed a Packet Mail reader system that implements many of the sophisticated features seen in other mail systems. This same program can also gateway mail between Packet Mail and Microsoft Mail for the Macintosh.

INTRODUCTION

Electronic mail systems on Local Area Networks (LANs) have evolved to the point where they are extremely robust and have many advanced features. These E-mail systems also usually have a quite advanced user interface, especially the ones that run on the Macintosh, MS-Windows and other windowing environments. There are quite advanced

news reader systems available for reading the bulletins on the Internet. Compuserve has sophisticated software for accessing its services. Many other mail systems have fancy software to make reading mail easier. The Amateur Radio Packet community needs software with this type of LOCAL INTELLIGENCE to make the process of getting and reading one's mail easier.

With fancy computers on our desks and the availability of sophisticated programs and networks, we should be able to have systems where our electronic mail comes to us. This is the way it is on many LAN based E-Mail systems. If you use Compuserve and the Compuserve Information Navigator, you also get this kind of luxury.

BRIEF HISTORY OF AUTOMATIC SYSTEMS FOR PACKET RADIO

In the early days of Packet Mail, the ONLY way to read messages and bulletins was to log onto your local PBBS and manually read them. This, unfortunately was (and still is) a slow process, especially at 1200 baud, and it is even worse if you are on a busy channel. Users wanted an easier way to read their packet mail.

Lan-Link for the PC has the capability to watch your local PBBS, detect if you have mail, connect to it when it sees mail for you, download the mail, and then disconnect. The messages are stored locally for you to access later at your convenience. Lan-Link can also read bulletins, and has a scripting system so that the user can customize it to his tastes and to be compatible with the PBBS system he is using.

The current generation of Packet Radio TNC's have 'mini' PBBS's built right into the TNC. PacComm calls theirs a PMS, (Personal Mail System), AEA calls theirs an MBX, (personal Mail Box), and the other manufacturers have similar features. These mini-PBBS's allow your local PBBS to forward your mail to you even when you are not there. Some sysops allow this and others do not.

Some people are also running stripped down versions of standard PBBS software, such as PRMBS and others as a 'personal' PBBS so that they can have their mail automatically forwarded to them.

All of the above solutions work and get the job done, some better than others. These types of solutions have their place, but most are designed just for personal messages and cannot handle large amounts of traffic.

The personal mail boxes work great if you and a few friends want to exchange mail but do not have good access to a full-function Packet BBS. This has the problem that they will only hold a very small amount of messages, therefore, are usually restricted to personal messages and not bulletins. This solution is generally used in less populated areas where full-function PBBS's aren't very common.

Lan-Link and the stripped-down versions of the PBBS systems solve the problem of not having enough space but require the full time use of the computer and do not allow anything else to be done at the same time.

The big problem with all of the above solutions is that you still end up with the same old command-line interface. This leaves the user with having to look through a single list of messages that were received

by his 'personal BBS system'. He still has to decide what to read and what not to. With some of the stripped-down PBBS systems, you can program them to select what types of bulletins you want to receive and what not to receive. Even so, the user still has a single list of messages to read through.

ADVANTAGES OF AUTOMATIC SYSTEMS

The main advantage to this type of system is that your computer can go get your messages for you. Therefore, this can happen when the frequency is less busy, or if it takes awhile, you are not sitting there waiting for it.

Since the sending and receiving is automatic, it can be scheduled to be done during non-prime time, i.e. late at night, or during normal work hours when there isn't much packet activity. This alone will improve the throughput for everybody, including the people that still read Packet Mail the old way because there will be less people on when they are on.

When you decide to read your messages, you can read them as fast as your computer can bring them up on your screen.

DISADVANTAGES OF AUTOMATIC SYSTEMS

Most of the automatic systems need to be left on 24 hours a day and require dedicating a computer to them.

Unless you are only receiving messages addressed directly to you, the system will occasionally receive messages that you didn't really want to read, thus causing more traffic than there would have been if you read your mail the old manual way.

If you are not careful, your hard disk will fill up very fast, especially when you are away from home for awhile.

INTELLIGENT AUTOMATIC SYSTEMS

We have already discussed Automatic systems above. I define an INTELLIGENT system as a system that can be tailored to get the types of messages that we want and not bother us with messages that we don't want. This has two main advantages. First, we are not bothered with things we are not interested in. Second, and more important, the airwaves are not wasted with the transfer of data that is not needed.

Given an automatic system for sending and receiving Packet Mail messages, you have the basis for developing a sophisticated user interface that can make the task of sorting through and reading the large amount of messages and bulletins significantly easier. With today's large volume of information, this type of electronic assistant is greatly needed. We already have an information overload which Packet bulletins are contributing to this also.

An intelligent system should be able to do many things for you. It should be able to

- Sort messages by topic.
- Scan subjects for keywords.
- Sort and select bulletins based on your personal preferences.
- Keep track of what messages you have read.
- Know about certain types of messages that are time critical.

- Update regularly sent messages such as satellite keplerian data sets.
- Keep track of the different people you send mail to so that you don't have to remember their full address every time you need to send them a message.
- Handle replying and forwarding of messages.
- Handle address lists and many other features that are common in LAN based E-Mail systems.

DESIGN GOALS

The following features are available on the LAN-based E-Mail system I use at work. I would like a system that does this for Packet Mail too:

- Automatic marking of read messages.
- Automatic address book.
- Easy scheduling.
- Easy replies and forwarding.
- Able to be run in the background.
- Address Lists.
- Aliases.
- Immediate notification when new messages arrive.
- User specifications of sorting criteria.

For a Packet Mail system the following features would also be desirable:

- Automatic sorting of bulletins by topic.
- 24 hour operation is not required.
- Easy selection of accept/reject criteria, based on either the TO/Distribution field, and the FROM field.
- Automatic superseding of messages such as KEPS.
- Selective automatic deleting of dated messages.

Having a system that implements even a few of these features would be a good step in the right direction. Also, with some PBBS's having over a hundred new messages a day, any system that can intelligently reduce message traffic is a plus for the whole system.

HISTORY OF OUR PROJECT

This project started out as a simple desire to write a gateway between Microsoft Mail for the Macintosh and the standard W0RLI/WA7MBL compatible PBBS systems. After pursuing this for awhile and getting some of it working, it was determined that by the time this project was done, we would have a message handling program that would have lots of nice features. These features would also be useful to someone that did not have Microsoft Mail. At this point, we backtracked a little and started working on a mail reader program that would also have the ability to forward messages to Microsoft Mail. This was implemented as an option that could be enabled or disabled.

With this type of gateway capability, we started to look at many other options too. The software is implemented such that it would be relatively easy to add gateways to other E-Mail systems and have them enabled or disabled also.

FUTURE CONSIDERATIONS FOR AUTOMATIC MAIL SYSTEMS

As discussed here, all of the above automatic mail systems work with existing PBBS protocols. The most common one in use today is the W0RLI/WA7MBL. There is growing support for the FBB compression protocol. Unless somebody

comes out with a better compression standard, we expect to implement this into our software.

As these automatic message systems become more and more widespread, we are going to start to see many different problems.

The biggest single problem that needs to be addressed is the ability to have a MULTICAST message system. A multicast message system is where a single PBBS will transmit a bulletin or message and as many other PBBS's that can hear it will receive it at the same time. This topic has been discussed several times in the past. See the references, [1], [2], and [3].

Another feature needed by mail protocols are remotely expanding mail lists, i.e. RMailer in the PRMBS PBBS system. This is the topic of a paper by Frank Warren, KB4CYC, also in these proceedings.

We also need to look into either expanding our current mail protocols or adopting others so that the Packet Mail systems can expand with our growing needs. [4]. Other features that would be useful would be return-receipt, enclosures, expiration dates, message size, supersede flags, i.e. this messages supersedes a previous message, and many other features not currently available.

These types of features are needed in the mail-forwarding protocol because the software is going to very rapidly need them. If they are not there, there will be lots of unneeded traffic that these additions to the protocols could avoid.

MACINTOSH PACKET MAIL PROTOCOLS

Mac Packet Mail, MPM, talks to other BBS's via the standard W0RLI/ WA7MBL protocols. The information exchanged in the header information of this protocol is enough for a PBBS system to know if it wants a particular message BEFORE it has actually received the entire message. In the actual protocol, the sending PBBS sends the header information, which includes what type of message, private or bulletin, who it is to, who it is from, and a BID/MID number. The TO address is either an actual person, or a distribution such as ALLBBS, or 4SALE, etc. MPM has a list of what types of messages it wants to receive. It also has a list of what types to reject. This is needed because there will always be people that address their messages incorrectly. Since the protocol allows for a message to be either rejected, received, or deferred, an intelligent system can also save the headers of other messages, and if the user wants to receive them, he can mark them for later retrieval. The system will get it the next time it logs on. This would be used for topics that the user usually doesn't want to read, but knows that once in a while he will want to. However, this would have to be set up ahead of time. Unfortunately, the subject is NOT transmitted with the header. It is not received unless the receiving BBS has said it wants the entire message.

MACINTOSH PACKET MAIL USER INTERFACE

The user interface for Macintosh Packet Mail has many features that resemble current LAN-based E-Mail systems. This interface is still under development as we continue to work on

the system (See **Figure 1**). This interface is what the end user sees and this is what will make or break a system like this. The goal is to make it easy for the user to navigate through the large piles of mail and bulletins that come through the system every day.

Once a user has this set up, he should be able to read his mail in much less time, get more out of it, and not have to wait for slow channels. This will do a lot to help the distribution of the vast quantities of information we have to deal with every day.

MACINTOSH PACKET MAIL MICROSOFT MAIL GATEWAY

Microsoft has a Software Development Kit available for developers that want to write gateways to Microsoft Mail. We purchased this gateway and used it to write the hooks into Microsoft Mail. The software supplied with this kit made developing a gateway into and out of Microsoft Mail easier than expected.

We had to develop our program so that it could read and write files that the Microsoft Mail gateway program could understand. This turned out to be pretty straightforward.

On the Macintosh, you can run multiple programs simultaneously, therefore, the Mail server and the Macintosh Packet Mail Program can run on the same computer. This allows one computer that is already running 24 hours a day to do the additional task of gatewaying between the LAN-based E-Mail system and Packet Mail.

Built into Microsoft Mail is an account/password system. The gateway system has an additional account system. This turned out to be very useful in restricting non-Hams from using the gateway by accident. All that has to be done is to create a gateway account for each Ham on the mail system and name the account with the user's call letters. With this method, even though the user's Microsoft Mail account is his name, the gateway account can be his call letters. This lets the gateway do the call letters to user mapping without having to write any extra software.

If someone on the mail system tries to use the gateway, he will get a rejection message telling him that he is not authorized to use that gateway and to contact his network administrator.

With this method of mail gateways, a user can send a single message to multiple people on the Microsoft Mail server AND to other people on Packet Mail.

The gateway software provides for convenient mail addressing too. Also, once you entered the call letters and BBS name once, you won't ever have to enter it again. See Figure 1 for who the BBS addressing is done.

CONCLUSIONS

With the onslaught of the information revolution, we are inundated with huge amounts of data. This is now starting to hit the Packet Mail systems. Therefore, to be able to keep up with what is going on, we need our computers to be able to do more work for us. Having a mail reader program that can make our reading of mail easier and quicker is very desirable.

In pursuing this type of sophisticated user interface, we discover that some things are not doable with the current protocols. Even though the message

transfer protocols that are used by the Packet Mail systems aren't very elaborate, it is still possible to develop a sophisticated package for the end user.

Figure 1

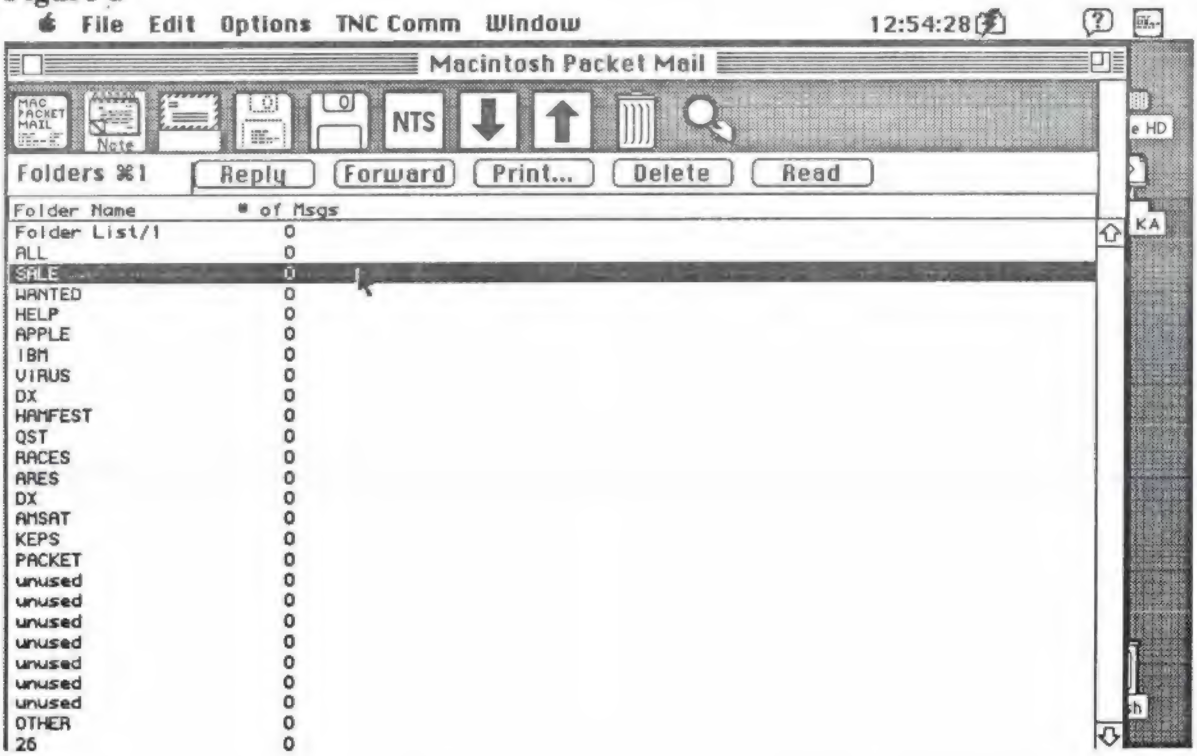
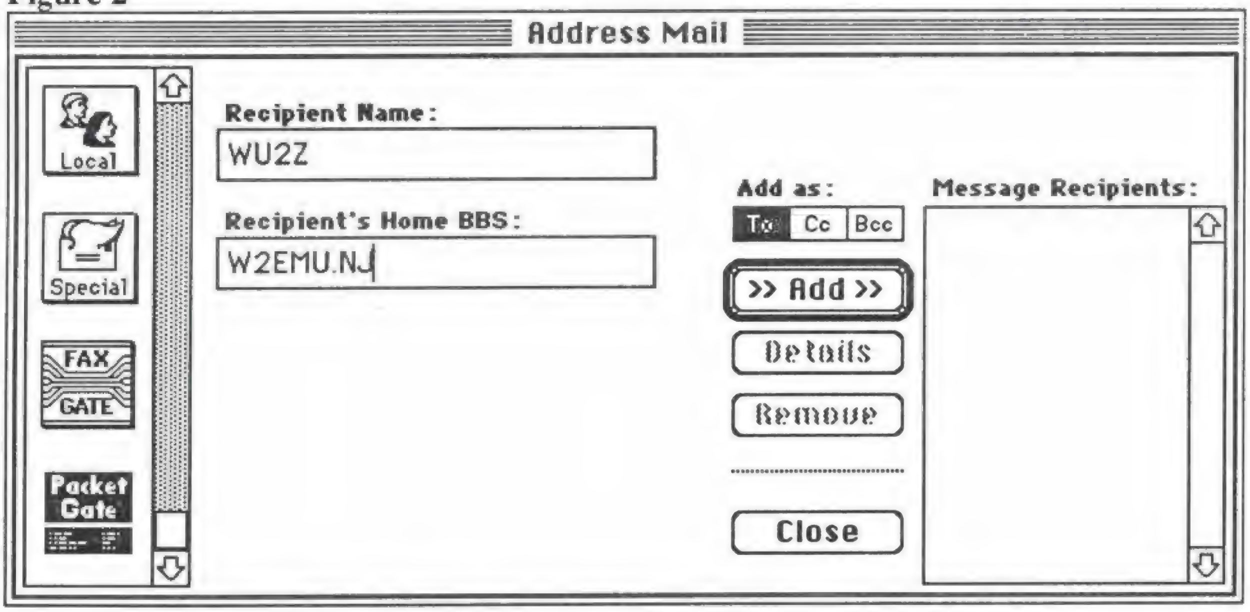


Figure 2



REFERENCES

[1] **Beattie, J. Gordon Jr.,** N2DSY *A Packet Broadcast Protocol,* Proceedings of the 8th Computer Networking Conference, Newington, CT: ARRL, 1989 pp 1-15

[2] **Koopman, Derek,** G1TLH *Connectionless Mail Protocol and More,* QEX ARRL Experimenters' Exchange, Newington, CT; ARRL, July 1990, pp 19-22

[3] **Price, Harold E.,** NK6K and **Ward, Jeff,** G0/K8KA *PACSAT Broadcast Protocol,* Proceedings of the 9th Computer Networking Conference, Newington, CT; ARRL, 1990, pp 232-244

[4] **Klemets, Anders,** SM0RGV *The Network News Transfer Protocol,* Proceedings of the 9th Computer Networking Conference, Newington, CT; ARRL, 1990, pp 150-153

HAPN-2: A Digital Multi-mode Controller for the IBM PC

John Vanden Berg, VE3DVV

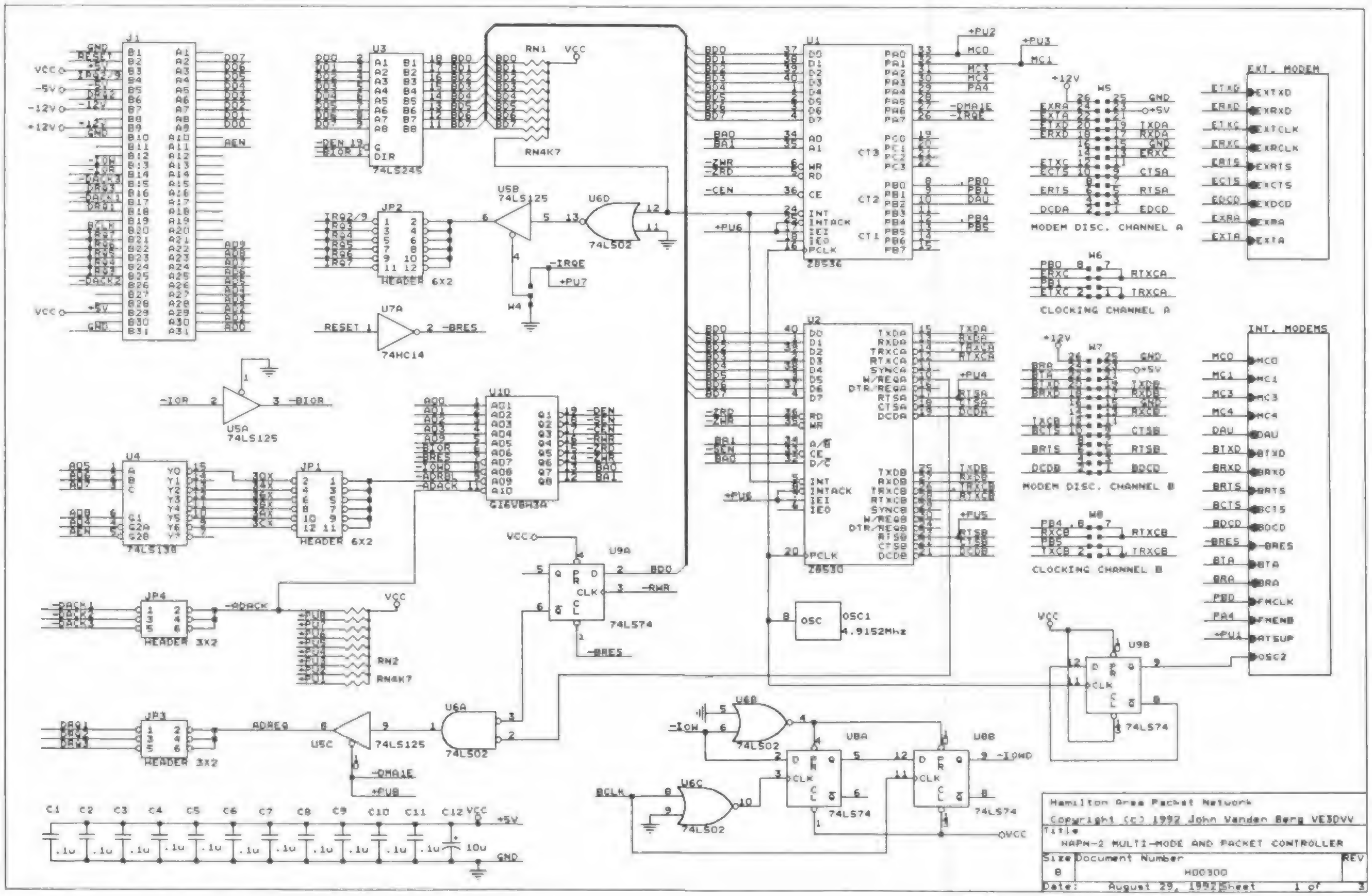
This paper describes a universal controller card for the IBM-PC bus. This system consists of one (or more) dual channel HAPN-2 adapter cards for packet radio. The basic card also has circuitry for experimenting with other modes such as RTTY, AMTOR, WEFAX, CW and SSTV.

Introduction

In 1985 we announced the first plug-in PACKET RADIO TNC for the IBM-PC (see Ham Radio Magazine, August 1986). The hardware and software were designed by the "Hamilton and Area Packet Network" (short HAPN) club, a group of experimenters. This card was called the HAPN-1 and was based on the now obsolete Intel 8273 SDLC chip. The card had one channel with a 1200 baud modem and used interrupts for operation. The card was also used with the HAPN 4800 baud medium speed modem (Ham Radio Magazine, August 1988). The interrupt driven software is somewhat limited due to the interrupt latency of the PC and the reliance of the timer-tick interrupt on the PC. The low-level device driver software was interrupt driven and stayed resident in the PC, allowing the adapter to run in the background.

The new HAPN-2 card.

The new card is about the same size 8.6" x 4.24" (21.8 x 10.8 cm) as the HAPN-1 and has a small prototype area. It contains the ZILOG Z8530 SCC serial communications controller and Z8536 CIO (counter timer and parallel I/O unit). The board supports 2 I/O channels for low and medium speed interrupt driven software. In addition Channel A can also be used under DMA control for real high speed modems such as the 56 Kb GRAPES modem. Channel B contains a AM7910 modem chip for conventional 1200 and 300 Bps (Bell-202 and Bell-103) operation. Additional modems such as the 4800 and 9600 Bps can be accommodated by plugging in to one of the two modem disconnect headers piggy-back. The A Channel can also be connected to an external modem via a DB25 using TTL, RS232 or RS422 interfacing.



J1
U3
74LS245
RN1
RN4K7
U1
Z8536
EXT. MODEM
W5
MODEM DISC. CHANNEL A
W6
CLOCKING CHANNEL A
JP2
HEADER 6X2
U5B
74LS125
U6D
74LS02
U7A
74HC14
U2
Z8530
W7
MODEM DISC. CHANNEL B
W8
CLOCKING CHANNEL B
INT. MODEMS
U5A
74LS125
U10
U4
74LS138
JP1
HEADER 6X2
U9A
74LS74
JP4
HEADER 3X2
RN2
RN4K7
OSC1
4.9152Mhz
U9B
74LS74
JP3
HEADER 3X2
U5C
74LS125
U6A
74LS02
U6B
74LS02
U8A
74LS74
U8B
74LS74
U6C
74LS02
Hamilton Area Packet Network
Copyright (c) 1992 John Vanden Berg VE3DVV
Title
HAPN-2 MULTI-MODE AND PACKET CONTROLLER
Size B
Document Number H00300
REV
Date: August 29, 1992
Sheet 1 of 3

The ISA PC-bus interface

For the PC interface refer to fig.1, the main diagram. U2 is the dual channel Z8530 clocked at 4.9152 Mhz from an oscillator module. The maximum speed of operation depends on this clock as well as the operating mode. For synchronous NRZI encoded data such as is currently being used for packet radio, this works out to a theoretical data rate of about 180 Kbit/sec. These speeds require fast DMA operation from the host computer. Higher speeds are possible by increasing the oscillator clock frequency and using external transmit/receive clocking. These high speeds rely on a short interrupt latency unless the CMOS version of the Z8530 is used. This chip contains more internal buffering for data and interrupts. A more practical expected speed for Radio packet is around 100 Kbit/sec on an ordinary 4.7 Mhz PC.

Chip U3 (74LS245) is a data buffer between the ISA bus and the adapter's internal data bus, and reduces the loading on the PC-bus to only one LS load. The load of any one signal line on the PC-bus is only one gate. This loading is important when the user has all his slots in the PC filled up with cards. SIP resistors RN1 provide the pull-ups for the internal data-bus. The adapter's address block is decoded with U4. The addresses of 30X, 34X, 36X, 38X, 3AX and 3CX are selections with JP1. The output goes to U10 a PAL or GAL that does the remainder of the address decoding as well as controlling the DMA signal. U5B gates the interrupt request line to the bus. The card needs its own IRQ line and JP2 is plugged for the proper IRQ. Note that U5B can float the IRQ line if W4 is moved to enable the IRQE line. This can be handy if the IRQ is shared with a COM port (irq 3 or 4). The IRQ line would be floating after a reset and only be enabled when the software initializes the adapter (activate IRQE line). The diagram shows this feature disabled so the hardware would be compatible with existing DRSI PC-Packet drivers which do not support this feature. This same floating feature is used with the DMA lines DRQ1-3. Jumper JP3 selects the DMA request and JP4 the DMA acknowledge lines. Both jumpers have to be in the same position for the DMA to work. The DMA can also be quickly enabled/disabled by the DMA enable flip-flop U9A. This is done for short intervals during programming of the SCC. The dual D flip-flops U8A and U8B delay the leading edge of the IO write signal required for the NMOS version of the Z8530. Both the Z8530 and the Z8536 CIO can generate interrupts to the system.

The Z8536 CIO

This chip contains 3 timers and a number of input/output lines. The I/O lines are used for enabling the DMA and IRQ signals enabling the adapter. PA0-3 are output lines and are used for the AM7910 modem. PA4 also going to the modem diagram (FMENB) selects between the AM7910 and the sine wave synthesizer to provide the transmit signal. The timers can be used in many different ways. When using the HAPNDRSI driver CT1 and

CT2 are used as divide by 32 for the transmit clock for the Z8530 SCC. The CT3 timer is used to provide TX-delay and TAIL-delay for the packets. When running the DRSI NOS driver CT3 is used to generate 10 Msec interrupts for polling the SCC. Using the high speed HAPN-2 NOS driver CT1 is used for TX-delay and TAIL-delay for SCC DMA channel A with 1 msec resolution. CT2 is used in 10 Msec resolution for the slower channel. In other than packet modes such as WEFAX, RTTY, CW, etc. CT2 is used for generating the transmit signal and also for decoding of the receive signal.

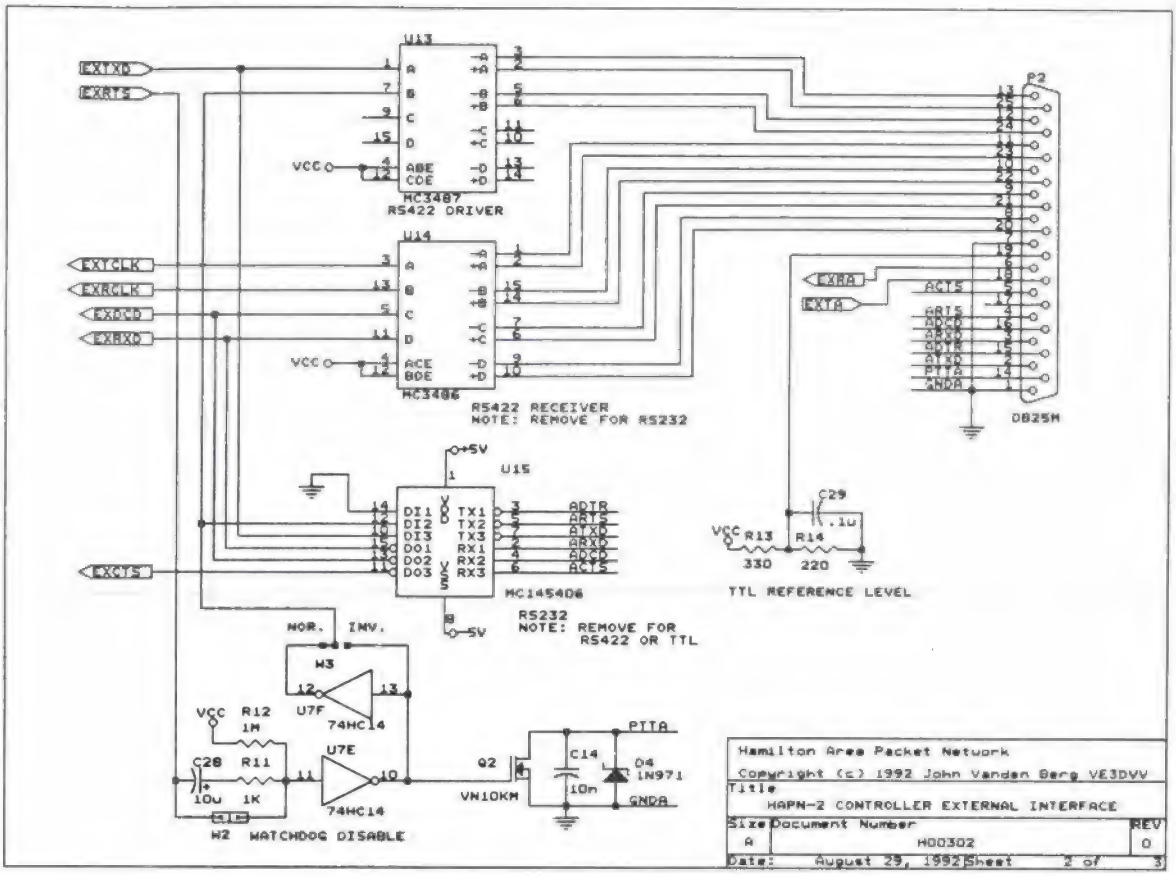
In short, the timers are handy to have around.

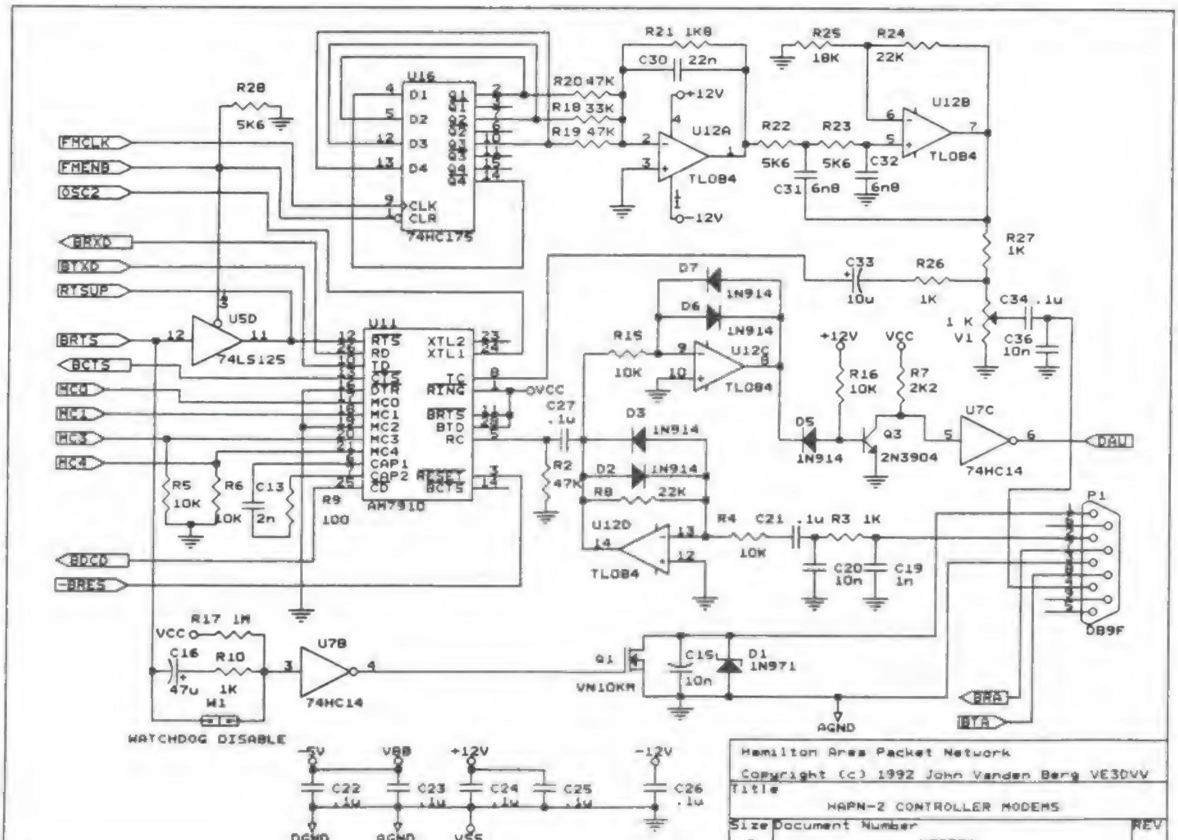
The Z8530 SCC

There are 2 versions of this chip, the basic NMOS version and the CMOS version. In this application we used the 6Mhz NMOS versions of the SCC and CIO chip. The CMOS version is really a better chip running faster, having more internal buffering and using less power, but is a bit more costly and not really required since the PCLK is only 4.9152 Mhz. The Cycle Recovering time, referring to the time period between any Read or Write cycles of the SCC is 6 PCLK cycles or about 1.2 microseconds. An input instruction on a basic 8086 takes 10 clock cycles, which on a 4.7Mhz XT (slowest PC I know of) works out to 2.1 microseconds. Considering other instructions usually used in a sequence, the SCC is certainly fast enough; however in fast 386 and 486 machines this Cycle Recovering time is important. Programmers have to put either a fixed or a variable delay in between successive I/O instructions. The variable delay would be determined by the programmer when actually measuring the speed of the computer.

The two channels on the SCC are independent of each other, and both have the same capacity when running in interrupt driven mode. The W/REQ line on channel A is brought out to allow this channel to work in DMA mode also. The HS modem is assumed to work half duplex in this mode. If more then one HS channel is required a second card can be added to the system. The A channel has no built-in modem, but supports an external modem either on the modem disconnect header W5 or via the DB25 connector on the back of the card. Selection of modem clocking between internal or external is done with jumpers on W6.

Channel B is the conventional slow speed packet channel for 1200 and 300 baud. However a medium speed modem could be substituted by installing the disconnect header W7. Control of clocking is done with W8.


MC3487
RS422 DRIVER
MC3486
RS422 RECEIVER
NOTE: REMOVE FOR RS232
MC145406
RS232
NOTE: REMOVE FOR RS422 OR TTL
TTL REFERENCE LEVEL
WATCHDOG DISABLE
DB25M
Hamilton Area Packet Network
Copyright (c) 1992 John Vanden Berg VE3DVV
Title
HAPN-2 CONTROLLER EXTERNAL INTERFACE
Size A
Document Number H00302
REV 0
Date: August 29, 1992
Sheet 2 of 3


74HC175
AM7910
WATCHDOG DISABLE
DB9F
Hamilton Area Packet Network
Copyright (c) 1992 John Vanden Berg VE3DVV
Title
HAPN-2 CONTROLLER MODEMS
Size A
Document Number H00301
Date: August 29, 1992
Sheet 3 of 3

External modem interface for channel A

We have two options. The first is to use the modem disconnect header W5 with a standard plug-in modem such as the 4800 and 9600 boards. The disconnect header includes the additional connections for power and audio in and out lines if one decides to use them (W5 pins 21 to 26). The second option is to use the DB25 connector for a modem outside the computer.

Refer to figure 2 for hook-ups via the DB25 connector on the card.
Schmidt trigger U7E with C28 and R12 produces a watchdog with a time delay of about 10 seconds. It can be bypassed with a jumper on W2 if required. Q2 is the driver for an open collector PTT line.

Selection between TTL, RS232 and RS422 is done by plugging in the appropriate chips. For RS232 levels only U15 (MC145406) is plugged in and U13 and U14 are removed.
The RS422 mode recommended for high speed is a TTL=like differential driver capable of driving long lines at high speed. Each signal requires 2 wires. A regular RS232 modem cable with all 25 wires can be used for hook-up. In this mode the RS232 chip U15 is removed.
For a TTL interface one could remove the interface chips altogether and plug in dip headers, bypassing the drivers. This is not recommended since the Z8530 would not be protected from the interface and the drive capacity is very limited. A better way is to use only one half of the RS422 signals since the levels are TTL compatible. By using either the + or - inputs and outputs one can invert the signal between negative or positive logic true levels. R13/R14 provide a bias for TTL reference. In TTL mode the floating inputs of U14 have to be connected (easily done in the DB25 connector) to this reference level at pin 19.

Built-in packet modem on channel B

Refer to fig. 3 for the built-in modems. U11 is the AM7910 modem chip providing 1200 baud (BELL-202) and 300 baud (BELL-103). A third mode for 600 baud (CCITT V.23) could also be selected using the MC0, MC1, MC3 and MC4 control lines via the CIO. The clock for the modem chip is derived from the main oscillator module by dividing this frequency by two with U9b (main diagram). U12D provides soft input limiting with D2 and D3 for protection of the AM7910. The gain of the op-amp is set at about two with R8 and R4. Prefiltering of the input is done with C19, R3, C20 and C21. It is interesting to note that the AM7910 also has an equalizer circuit for 1200 baud that can be selected via the control signals.

Here is the AM7910 selection table:

MC4	MC3	MC2	MC1	MC0	
0	0	0	1	0	Bell 202 1200 Bps half duplex tones 2200 1200 Hz
0	0	0	1	1	Bell 202 1200 Bps + equalizer half duplex tones 2200 1200Hz
0	1	0	0	0	CCITV V.23 Mode 600 Bps half duplex tones 1700 1300Hz
1	0	0	0	0	Bell 103 300 Bps Orig loopback tones 1070 1270Hz
1	0	0	0	1	Bell 103 300 Bps Ans loopback tones 2025 2225Hz

Note : For a complete list of modes refer to the AM7910 Technical manual available from Advanced Micro Devices.

Chip U7B (74LS14) is a hex schmidt trigger used as a watchdog. The time constant R17/C16 make it kick in after about 30 seconds. By putting a jumper on W1 the watchdog can be bypassed. Q1 drives the PTT. The modems output is adjustable with V1 between 0 and .6Vpp.

Modem for other then packet modes

Op-amp U12C will hard limit the receive signal and Q3 with D5 converts it to a TTL level by switching at the zero crossing. The schmidt trigger U7C shapes it to a fast rising pulse. The output goes to the trigger input of CT2 of the Z8536. This circuit can be used for frequency measurement and demodulating CW, RTTY, AMTOR, SSTV, WEFAX etc. The sending part for these modes consists of the quad D flip-flop U16 hooked up in a ring, driving three resistors of the digital to analog converter (R18, R19, R20 and op-amp U12A) to generate a simple stair sine wave. The signal is filtered by U12B to produce a clean sine wave. The frequency of the sine wave is directly proportional to the clock frequency at pin 9 of U16. This signal (FMCLK) comes from the output of the Z8536 timer CT2. All that is required to produce an audio sine wave is to have the timer produce a square wave of 8 times the required audio frequency. The selection between packet and other modes is done by software, controlling the reset of the ring counter with FMENB and enabling the RTS for the AM7910 modem.

Conclusion

The design was made for the interest of the packeteer who looks forward for higher packet speeds and at the same time wants to enjoy the conventional 1200 Bps mode. Also it will let him experiment with other radio modes (CW, RTTY, WEFAX and SSTV) as well. These modes were added with little additional cost since most of the hardware was already present. TCP/IP networking at high and low speed are practical using NOS. The higher level software we had developed for the earlier HAPN-1 adapter can be used on the HAPN-2 as well due to the modular design and substitution of a different device driver. The board is available from HAPN in either assembled and tested form or in KIT form. For details contact H.A.P.N.

Note: The address of HAPN has changed!!

The new address is:

H.A.P.N.
5193 White Church Rd.
Mount Hope, Ontario
L0R 1W0 CANADA

Acknowledgements

I wish to thank Jack Botner (VE3LNY) for his software contributions and Max Pizzolato (VE3DNM) with the design of the circuit board.

References

[1] Heatherington, Dale, WA4DSY "A 56 kilobaud RF Modem"
Proceedings of the 6th ARRL Computer Networking Conference
Newington CT: ARRL 1987

[2] Botner, Jack, VE3LNY, Pizzolato, Max, VE3DNM, John Vanden
Berg, VE3DVV "A PACKET RADIO TNC FOR THE IBM PC"
HAM RADIO: August 1986

NOSVIEW

The On-Line Documentation Package for NOS

by Ian Wade

G3NRW @ GB7BIL.#27.GBR.EU
[44.131.5.2]
uad1200 @ dircon.co.uk

NOSVIEW is an on-line documentation package for the KA9Q Network Operating System (NOS). First released in September 1991, NOSVIEW is a complete reference work describing in detail all of the commands to be found in the major NOS releases. This paper outlines its main features, and how to get a copy.

Introducing NOSVIEW

Over the years, many documents have appeared on the networks describing various features of NOS, but much of that material is incomplete. Some of it is inaccurate, and, because it was written and edited by many hands, sometimes very misleading and inconsistent.

My small contribution to the genre is NOSVIEW. In NOSVIEW I have attempted to pull together all the available documentation and massage it into a consistent whole.

The final product is almost 300 pages long, around 20 percent being new material. All of the NOS commands are described in detail, and there is at least one example included with each command. There are also many examples of display outputs, showing the results of executing the commands.

Consistency

Because NOS contains software modules originating from several different sources, the associated documentation inevitably contains inconsistencies.

For example, the words *label* and *interface* apparently describe different objects, whereas in actuality they are the same thing. On the other hand, the word *address* can have different meanings, depending on the command.

A lot of effort has gone into NOSVIEW to eliminate these inconsistencies. Command parameter names are now consistent throughout. Callsigns in the examples follow a set pattern: calls for NOS stations are in the *NS9xxx* series, vanilla AX.25 stations are *AX9xxx*, NET/ROM stations are *NR9xxx*, and so on.

Also, to distinguish between IP hostnames and AX.25 callsigns, hostnames are shown in lower case (*ns9abc*), whereas AX.25 callsigns are in upper case (*NS9ABC-5*).

These seemingly simple rules make a tremendous difference to the readability of the documentation. There is now no doubt about whether a parameter should be an IP hostname or an AX.25 callsign, or whether you need an IRQ number or an interrupt vector address, and so on.

NOSVIEW On-line

But this is only half the story. The real power of NOSVIEW comes into its own when used with VIEW, a public domain file viewer included with NOSVIEW. VIEW lets you hot-key to the NOSVIEW documentation without breaking out of NOS, providing instant on-line help whenever you need to know what to do next.

Figure 1 shows an example of the VIEW screen. To take full advantage of VIEW, NOSVIEW is supplied as a set of over 90 individual help files, one file for each NOS command. This provides immediate access to the command of interest, saving time and effort when searching for detailed information.

A further benefit of supplying NOSVIEW as individual files rather than one monolithic document is that you can place the files in your NOS public directory. Then when someone logs into your system, they can download selected NOSVIEW information in manageable pieces, rather than saturate the network for hours on end trying to download one enormous file.

```
Load file:  V:*.*                                    ┌── FILES ──┐
                                                      NRSTAT
              route add default tnc0                  PARAM
              route add ns9jim tnc0                   PING
              route add 192.3.4.5 sl0   (no gateway   POP
                                         point-to-p   PPP
                                                      PS
route addprivate <target_host>[/bits] | default <i    PWD
                    [<gateway_host> [<metric>]        RARP
                                                      RECORD
The 'route addprivate' command is identical to 'ro    REMOTE
that it also marks the new entry as private; it wi    RENAME
included in outgoing RIP updates.                     RESET
                                                      RIP
>> Example:  route addprivate region41/24 ns9bob      RLOGIN
                                                      RMDIR
                                                      ROUTE
                                                      RSPF
route drop <target_host>[/bits]                       SCCSTAT
                                                      SESSION
The 'route drop' command deletes an entry from the    SHELL
packet arrives for the deleted address and a defau    SMTP
effect, it will be used.                              SOCKET

↑ ↓ PgDn PgUp End Home Esc F3/Load F1/? help Action Colours Graphics Hex Wrap
```

Fig 1: On-line NOS documentation with NOSVIEW.

There is a separate help file for each NOS command, selected from the pop-up FILES menu.

The NOS Get-Away Special

Yet another feature of NOSVIEW is that it contains a complete working set of NOS software, dubbed NOSGAS — the NOS Get-Away Special. NOSGAS incorporates a complete set of supporting files (such as *AUTOEXEC.NOS*, *FTPUSERS* and so on) which you can use on your system. The templates are accompanied by full descriptions of their formats, plus warnings about the "gotchas" which can cause lots of frustration if you are unaware of them.

All you have to do is edit these templates to match your system (by modifying callsigns, etc), and you have a ready-made environment to try out NOS.

How to get NOSVIEW

The latest release of NOSVIEW is version 244; i.e. released in 1992, week 44. By now, NOSVIEW should be available on the major telephone bulletin boards worldwide, and also on *ucsd.edu* in directory *hamradio/packet/tcpip/docs*. Look for the files *NOSVIEW.ZIP* and *NOSVW244.ZIP*.

Alternatively, you can get a free copy by mailing a DOS-formatted diskette (any size *except* 360K) and return mailer to: Ian Wade, 7 Daubeney Close, Harlington, Dunstable, Beds, LU5 6NF, United Kingdom.

Please enclose return postage as follows:

United Kingdom:	UK postage stamps
Europe:	3 IRCs
The Americas:	7 IRCs
Rest of World:	9 IRCs

(Any unused IRCs will of course be returned).

NOSINTRO: The book

NOSVIEW is an advanced reference document intended mainly for people who have already got NOS up and running. Beginners will probably find it heavy going, but may be interested to know that my new book on NOS — *NOSINTRO* — is (at last!) ready. Full order details are included within the NOSVIEW package.

Network Enhancements Implemented in the CT/NJ/NY Region

Frank Warren, Jr., KB4CYC
Andrew Funk, KB7UV
Scott Weiss, KB2EAR

Ad-Hoc Tri-State Managed Packet Group (@MGTBBS)

The problems of the prolifferation of flood routings, widespread mesh forwarding and an ever-expanding system census had combined to reach a point where the PBBS network in the CT/NJ/NY tri-state region was in dire need of an overhaul. This paper details the approaches taken by the majority of systems in the region to address these problems.

1. Introduction

As part of the June 1992 ARRL Hudson Division Convention, a forum was held for regional packet radio System Operators (sysops) and Network Administrators. Our aim was to begin dialog among those operating packet systems in the region, with the goal of improving the packet environment in the region for all concerned. (The systems "talked" with each other, but not the people behind the systems... Until this meeting.)

Since this June meeting, Sysops and Network Managers in the tri-state region have continued meeting and planing. This work has developed a cooperative system for the distribution of bulletins. The solution developed combines a consensus on which distribution routes will be supported, a list of suggested "To:" fields which users are encourged to use, cellular hub and spokes bulletin distribution topology, and time

reserved exclusively for user access to PBBSs and other network services.

2. Forwarding 'Quiet Hours'

The initial decision reached by the group was to prohibit BBS-to-BBS forwarding between 1800 and 2400 local, daily, on all paths which may also carry real-time user data. This provides users with six hours each day, during "prime time," when their enjoyment of the packet network is not impaired by contending with automated stations.

3. Hub and Spokes Forwarding Topology

At the group's mid-July meeting, a plan to reduce bandwidth consumption by bulletin distribution was formulated. Bulletin distribution now follows a cellularized, hub/spoke or server/client design.

Many of the systems in the region use a series of regional backbone nodes maintained as part of the Eastnet Backbone Network (EBN). Others are served by the ROSE X.25 Packet Network maintained by the Radio Amateur Telecommunications Society (RATS).

Those systems using the EBN regional nodes receive their bulletins from a single, designated hub/server within their "cell." The cells were defined based upon the existing backbone EBN nodes. The cells currently resolve to Connecticut, Long Island, New York City, Downstate New York, Northern New Jersey and Central New Jersey. Cell size and definitions may change, over time, as a function of network traffic and topology.

The RATS ROSE Network has bi-directional connectivity with the two New Jersey EBN-based cells and a bi-directional feed to the cellularized (non-EBN) network in Southern New Jersey. Bulletin distribution for systems on this network also follows the client/server model.

This design has freed the bandwidth previously occupied (wasted!) by everyone trying to forward everything to everyone else.

In addition, this dual-network topology provides redundancy and robustness often lacking in Amateur Packet networks.

4. Supported Flood Distributions

The following is the list of flood distributions (@-field routes) the region has decided to support for forwarding:

Bulletin Flood Routes	
Route	**Description and Useage**
*xx*BBS	Distribution to areas other than a state (ARRL sections, etc.)
*xx*NET	State–wide routing, using 2–letter state designation *xx*
AMSAT	Amateur satellite (AMSAT) bulletins
ARESCT	Conecticut Amateur Radio Emergency Service bulletins
ARL	ARRL bulletins (@ARRL is *not* to be distributed)
ATLDIV	ARRL Atlantic Division distribution
CTBBS	Connecticut ARRL Section distribution (same as CTNET)
CTNET	State of Connecticut distribution (same as CT ARRL Section, CTBBS)
ENYBBS	ARRL ENY Section distribution
EPABBS	ARRL EPA Section distribution
HUDSON	ARRL Hudson Division distribution
LOCAL	Non–flood bulletin, for ONE LOCAL PBBS ONLY
MGTBBS	Administrative distribution for NJ/NY/CT Ad–Hoc Managed Network
NASA	Material for NASA sources
NEBBS	New England regional distribution (CT, MA, ME, NH, RI and VT)
NEWHDR	New Headers parsed by the N2MH program
NJEOC	NJ State Office of Emergency Management "Official" bulletins
NJNET	NJ state distribution
NJPSC	NJ Public Service Communications (includes ARES)
NLIBBS	ARRL NLI Section distribution
NNJBBS	ARRL NNJ Section distribution
NYNET	State of New York distribution
PANET	State of Pennsylvania distribution
SNJBBS	ARRL SNJ Section distribution
TRIBBS	Tri–State (CT, NY and NJ) regional distribution
USBBS	United States distribution (replaces ALLUS, ALLUSA, USABBS, USA)
WNYBBS	ARRL WNY Section distribution
WW	World–Wide distribution (replaces ALLBBS, WWW)

5. Suggested "To:" Fields

The following is a list of "To:" fields the group decided to distribute as a partial list of suggestions. The entries for the various PBBS software were originaly proposed as flood routes, but were recast as "To:" values based on explicit statements and examples from several PBBS software authors.

Suggested "To:" Field Useage	
"To:"	Useage
ALL	Should only be used if nothing else applies!
AMSAT	AMSAT–specific space/satellite information
BEACON	Beacon lists and information
CBBS	Program–related distribution: CBBS
CLASS	Amateur Radio and other class announcements
DX	DX related information and questions
EVENT	Special events, on–air or not, including hamfests
EXAM	VE Exam session announcements
FBB	Program–related distribution: FBB
HELP	Requests for help which don't fit into other categories
ICOM	Icom product–specific postings
KEPS	Keplerian elements (satellite tracking)
KENWD	Kenwwod product–specific postings
MBLBBS	Program–related distribution: MBL
MSYS	Program–related distribution: MSYS
PRMBS	Program–related distribution: ROSErver/PRMBS
PROP	Propagation reports
QSL	QSL information: routes, managers, etc.
REBBS	Program–related distribution: AA4RE
RLIBBS	Program–related distribution: RLI
SALE	Items for sale (Amateur Radio, of course!)
SWAP	Items offered for swap
SWL	Short Wave Listening
SYSOP	For System Operators (usually type "P")
USERS	Postings for System or Network Users
WANT	Items wanted
YAESU	Yaesu product–specific postings

6. Conclusion

The plan outlined above, combined with ongoing efforts in user education by the participating SYSOPs, has improved packet operation throughout this region. While not all of these steps may be as useful in other areas of the country, they may serve as a basis for development of a broad based (dare we hope world wide?) consensus. We also urge the adoption of dedicated user time, for without users our systems are not needed.

7. Contacting The Authors

The authors of this paper, along with the sysops of all systems participating in the Ad-Hoc Tri-State Managed Packet Group, can be contacted by sending (using the "SP" command) a single packet message addressed to:

RMAIL@KB4CYC.NJ.USA

and containing as the first line of text the following:

To: rmail@kb4cyc.nj.usa, sysop@mgtbbs

This Remote MAIL message will be processed automatically at the KB4CYC PBBS and become a flood bulletin to all the participating MGTBBS systems. (See the paper, "RMAILER: A Remote Ad Hoc Mailing List Expander," elsewhere in these procedings for a complete explanation of Remote MAIL.)

Alternatively, as each of the authors operates a PBBS, they may be reached via packet radio using the following addresses:

kb4cyc@kb4cyc.nj.usa
kb7uv@kb7uv.#nli.ny.usa
kb2ear@kb2ear.nj.usa

RMAILER: A Remote Ad Hoc Mailing List Expander

Frank Warren, Jr., KB4CYC
Andrew Funk, KB7UV

Radio Amateur Telecommunications Society

The ROSErver/PRMBS BBS system supports a remote ad hoc mailing list protocol known as RMAIL (Remote MAILer). ROSE/RMAILER provides RMAIL support for other BBS systems operating under MS-DOS*.

1. Background

The RMAIL protocol provides a mechanism similar to National Traffic System "book" traffic for the Amateur Radio store-and-forward BBS network. Both protocols are designed for situations where identical messages are sent to multiple recipients. A single copy of such a message, including addressing information for the intended recipients, will traverse the network to a point closer to final delivery. Upon reaching that point the single message will expand into multiple messages, one for each recipient.

Such protocols result in clearly measurable improvements in channel utilization. This improvement, for *n* recipients, approaches (n-1) times the single recipient message size.

While other approaches to mailing lists are centralized and depend upon external maintenance of such lists, RMAIL operates upon a list defined during message origination and carried within the message. RMAIL, therefore, imposes no additional clerical burdens upon packet BBS system operators.

Centralized mailing list servers can take advantage of the ad hoc mailing lists provided by RMAILer if they are "RMAIL aware." (At this time the only "RMAIL aware" centralized distribution server is ROSEDIST, an integral part of the ROSErver/PRMBS PBBS package by Brian Riley, KA2BQE.)

2. Required Message Elements

Two elements must be present for successful processing of a message by RMAILer. First, the message must be addressed to RMAIL@*<bbs_name>*, where *<bbs_name>* is the callsign or hierarchical name of the target BBS for RMAILer processing (message expansion). Second, the message must contain an RFC-822 style "To:" line of the form:

To: RMAIL@*<bbs_name>, call@bbs1, call@bbs2,...,calln@bbsn*

where *<bbs_name>* is again the callsign or hierarchical name of the target BBS for RMAILer processing, and *callx@bbsx* is replaced by the PBBS address of each individual recipient. (There is a 512 characters for the "To:" line; see "Current Limits and Future Plans" for discussion.)

These elements are automatically created and properly included in RMAIL messages originated on PBBS systems with integral RMAIL support. (Currently this feature is only available on ROSErver/PRMBS PBBSs.) Users of other PBBSs wishing to create RMAIL messages will need to manually create the required "To:" line as the first line of the message. Even when this results in multiple RFC-822 style "To:" lines RMAILer will find the data and successfully expand the message.

3. History

Several meetings of PBBS and packet network system operators (sysops) in the CT/NJ/NY region were held this summer. (Further details of these meetings appear elsewhere in these proceedings.) As a result of these meetings, all PBBS sysops will be exchanging information on a regular basis. While discussing the distribution of these information bulletins the advantages of RMAIL became apparent.

Understandably, sysops become "attached" to their program-of-choice, and are unlikely to change just to be able to utilize RMAIL. However, most PBBS software supports a standard format for file-based Import/Export of messages. In addition, most can also be configured to automatically run external

programs. Many "servers" already exist which provide enhanced capabilities to PBBSs by utilizing these techniques. Those non-ROSErver/PRMBS sysops in attendance expressed a desire for a similar server to handle RMAIL expansion.

This software, written by Frank Warren, KB4CYC, is the result.

4. Design and Operation

RMAILER is designed to be run as a server program from the event cycle (or equivalent) of the BBS. Messages to be processed by the program must first be exported to a file. Next RMAILer is run, creating or appending to an output file containing the expanded messages. Then the BBS imports the file containing these messages. Finally, the output file should be deleted to prevent future message duplication.

File names used by RMAILer may be specified on the command line. If not, RMAILER will look to process a file named `RMAILER.EXP` and append its output to `RMAILER.IMP`.

5. Current Limits and Future Plans

At this time the "To:" line is limited in length to 512 characters and RFC-822 style continuation lines are not supported. Plans are to address this by supporting RFC-822 continuation lines in a future release.

A companion centralized distribution list server, based upon the ROSErver/PRMBS ROSEDIST program, is contemplated.

6. Distribution

ROSE/RMAILER is available at no charge for non-commercial use within the Amateur Radio, MARS, RACES, and CAP services.

RMAILER is distributed as a self-extracting LHarc archive `RMAILER.COM` containing files `RMAILER.EXE` (the executable) and `RMAILER.MAN` (a UNIX™ style "manual page").

ROSE/RMAILER can be downloaded from CompuServe™ HamNet Forum, the KB7UV Landline ROSErver/PRMBS (see below), "HIRAM" (the ARRL multi-user

telephone BBS), and other telephone BBSs.

Those desiring RMAILer on MS-DOS magnetic media should send pre-formatted diskettes and return postage to the Radio Amateur Telecommunications Society (see below).

Requests for the code via the Internet should be directed to:
kb4cyc@kb2ear.ampr.org

Frank Warren, the program's author, may be contacted via packet as:
kb4cyc@kb4cyc.nj.usa

7. The RATS Open Systems Environment

ROSE/RMAILer is an element of the RATS Open Systems Environment (ROSE), a project of the Radio Amateur Telecommunications Society (RATS).

Other elements of ROSE include the ROSE X.25 Packet Switch by Tom Moulton, W2VY; the ROSE/OCS Online Callbook Server by Keith Sproul, WU2Z, and Mark Sproul, KB2ICI; ROSErver/PRMBS, the Packet Radio MailBox System by Brian Riley, KA2BQE; and ROSE/STS Station Traffic System by Frank Warren, KB4CYC.

Correspondence may be sent to:

RATS
PO Box 93
Park Ridge, NJ 07656-0093

The RATS KB7UV Landline ROSErver/PRMBS supports data rates of 1200 to 9600 bps (V.32), and J-, X-, Y-, and Zmodem binary protocols. It can be reached at 718-956-7133. Callers should wait for the "login:" prompt (don't even press ENTER!) and follow the instructions provided.

7. Acknowledgments

ROSE/RMAILer would not be possible were it not for the pioneering RMAIL development included within the ROSErver/PRMBS Packet Radio MailBox System package by Brian Riley, KA2BQE. We also thank the members of the Radio Amateur Telecommunications Society for their continued support.

SOFTKISS

TNC-Less Packet for the Macintosh

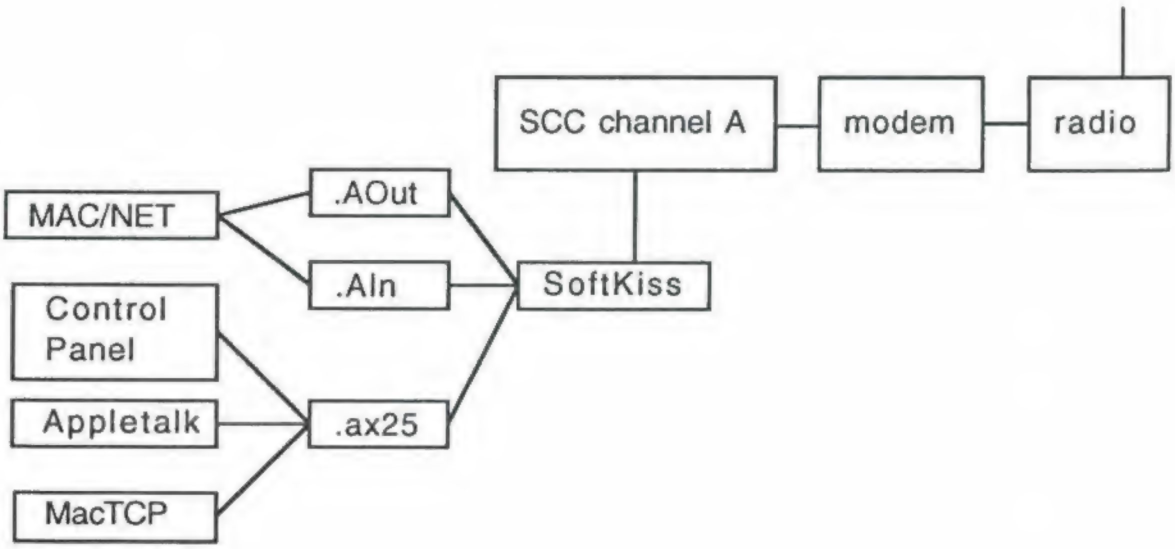
Aaron Wohl, N3LIW
6393 Penn Avenue, #303
Pittsburgh, PA 15206
n3liw+@cmu.edu (Internet)
n3liw (America Online)

ABSTRACT

This paper presents the design and implementation of a software TNC emulator for the Macintosh called SoftKiss. The tools used to build Softkiss are described. These tools will be useful for any Macintosh device driver development. Macintosh TNC less packet is compared to IBM PC TNC less packet.

INTRODUCTION

Softkiss is a family of drivers for the Macintosh. The Mac is connected to a RF modem. Softkiss emulates a TNC in kiss mode. The Mac serial drivers are replaced. Existing Mac programs access the serial port as if a tnc in kiss mode where attached.



COMPARISON WITH IBM PC TNC-LESS PACKET

IBM PC clones use various serial controller chips and line drivers. A typical IBM PC has control signals for at least DTR and carrier detect. However, some implementations such as the Hewlett Packard HP95 have a three wire implementation with no extra signals for transmitter control. The typical output drivers are capable of supplying enough current to power a CMOS modem. Although some systems use charge pumps to develop the RS232 +12 and -12 voltages and can not supply sufficient current to power a modem. The IBM PC serial ports can not directly handle the synchronous HDLC bitstream used in packet radio.

The "Mac" computer may be any one from the original 128K byte Mac to a quadra 950. All of the Macs have a Zilog 85C30 UART connected to high current drivers that can power a CMOS modem. The portable and power book models adjust power to various subsystems such as the serial port and also vary the processor clock speed to conserve power. Currently, when Softkiss is active it leaves the appropriate serial port powered up and disables the slower processor clock and sleep mode.

All of the Macs have high power data out, data in, and HSKo (handshake output, typically called DTR (data terminal ready)). HSKo is used by SoftKiss to key the transmitter. From there the hardware diverges. There are two input pins HSKi (commonly called carrier detect) and GPi (general purpose input). The HSKi is available on all Macs except the Macintosh LC where it is not connected.

The GPi varies considerably from Mac to Mac. It is nominally connected to the DCD input on the SCC chip. However on the Mac+ the mouse horizontal is connected to one SCC channel and the mouse vertical is connected to the other. On the Mac+ this interrupt must be left enabled for the mouse. On some Macs this DCD input on the SCC can be switched to GPi input or to a high speed clock line. The sense of setting this switch varies. The DCD interrupt enable bit in the SCC is write only. Softkiss needs to reset the SCC to set it up to run in HDLC mode. Currently, Softkiss guesses whether to enable the DCD interrupt based on checking the DCD interrupt vector to examine if it points to a return instruction.

On the newer high-end Macs the SCC chip is no longer directly connected to the main processor. An auxiliary IO processor (a Microchip PIC16C5x chip) usually handles serial IO. Apple supplies a "Compatibility Switch" control panel to disable the auxiliary processor. Softkiss examines the "HasSCC" gestalt selector to make a guess as to which mode the processor is in. If the SCC is accessed while the PIC chip is enabled then the system will hang. There is no published interface to tell if the PIC chip is enabled.

Some of the power book processors use an auxiliary processor for power management. When the OS code power management routine talks to the power manager processor it disables interrupts for long periods of time -- long enough to be a problem for MIDI input. On one of the serial channels there is a hook called while interrupts are off to collect incoming data bytes. But this hook is a hack and Apple doesn't recommend using it. Softkiss does not use the hook at this time. This has not been a problem at 1200 baud. However, hams intending to use highspeed packet (above 4800 baud) should beware.

Softkiss needs to remove the existing serial drivers for a port (.AOut and .AIn or .BOut and .Bin) and install itself. When Softkiss is closed it needs to replace the original drivers. This replace/restore is not supported by the OS. There is a remove driver call, but you can't tell which driver to put back (the serial driver may have come from ROM or the system file and it may have been patched to fix bugs at boot time). Softkiss directly manipulates the OS driver table to save and restore the old serial driver.

.AX25 driver

Softkiss installs a new driver .AX25. This driver is used to send control information to Softkiss itself. It will also be used to read and write packets for the interface to the Apple MacTCP and Appletalk.

MACINTOSH DRIVER WRITING TOOLS

dbo_printf - Screen output on the Macintosh must not be done from interrupt level as the drawing and heap management routines in the OS are not reentrant. Setting breakpoints in interrupt level code is problematic. In the past when the author really needed some information from a device driver he would dump the register to the memory mapped screen buffer and read the pixels with a magnifying glass. Fortunately, as part of Softkiss there is a debugging output routine. It writes directly to the screen memory (the screen must be in black and white 2 bit mode). dbo_printf may be called from any interrupt level and is reentrant.

driver_shell - Device drivers must have a unique driver number. Since there are only about 48 driver slots (depending on the OS version) fixed numbers are not issued outside of Apple. According to the Apple documentation drivers are opened with OpenDriver(). This will pick up the driver number stored in the resource fork (program). Attempts to dynamically set this number will trigger many of the virus prevention programs. The driver shell package that comes with Softkiss dynamically finds a free driver slot and installs a driver using the more obscure InstallDriver() call. Driver shell actually installs a stub which calls any desired code. Since Softkiss is reentrant the port A and the port B code share the same main code.

OVERALL CODE COMMENTS

Softkiss started out as a Macintosh application using no interrupts. I recommend this when starting out to access a new device. The THINK C high level debugger can be used to single step through code and examine variables. Interrupts can be added to the application. The core of the Softkiss driver can be linked into a test application for debugging or with the driver shell to make a stand-alone driver.

The source code is available from the author at no charge or from CompuServ hamnet library 9, or America Online in the Ham area.

ABOUT HIGH SPEED PACKET

Softkiss handles each interrupt individually. There is an interrupt when the sync character is detected and another when the first data byte comes in. For high speed, Softkiss should check after handling each interrupt if additional data is available. Appletalk spinloops at interrupt level to receive data at 250 kbits/s. Currently with a loopback cable, Softkiss can talk to itself at 4800 baud on a power book. The author has a pair of Kantronics D4-10 radios and intends to use the bit slicer for 19.2kb without a modem.

HIGHER LEVEL PROTOCOL STACKS

AX25

Currently MAC/NET, a port of KA9Qs net program by Tetherless Access Ltd., is used for the higher level protocol stacks. Jim Van Peursem, KE0PH, is working on an native Mac AX25 stack and terminal program called Savant.

IP/TCP

I have the interface specs to Apple's MacTCP package and will work on the interface glue to Softkiss as time allows.

NATIVE APPLETALK

Native Appletalk support would allow remote access to Macintosh file servers and printing via ham radio. I have the interface specs for adding Appletalk drivers. However, Appletalk uses dynamic node number assignment and it isn't clear how to do node number arbitration with the very dynamic nature of ham usage and marginal paths. Readers interested in helping to solve this (or any other) problem should contact the autho

References

[1] Francis, Dexter, et al "Packet on the Mac". 73 Amateur Radio. October, 1992, pp 8.

[2] Hendricks, Dewayne WA8DZP and Thom, Doug N6OYU. "KA9Q Internet Protocol Package on the Apple Macintosh". Proceedings of the 8th ARRL Computer Networking Conference. Newington CT: ARRL, 1989 pp. 83-91.

Notes

Notes

Notes

Published by: THE AMERICAN RADIO RELAY LEAGUE

Published by: THE AMERICAN RADIO RELAY LEAGUE